Public-Key Pseudorandom Codes from Distorted McEliece Assumptions

Victor Dyseryn[1], Danilo Francati[2], and Daniele Venturi[2]

[1] LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
victor.dyseryn@telecom-paris.fr
[2] Sapienza University of Rome
{francati,venturi}@di.uniroma1.it

Abstract. Pseudorandom codes (PRCs), introduced at Crypto 2024 by Christ and Gunn, are encryption schemes with pseudorandom ciphertexts and error-correction guarantees. PRCs are useful as a tool to obtain watermarking for generative models, in particular ensuring that a watermark is hard to remove against an attacker that can modify up to a given fraction of the watermarked output (a.k.a. the *robustness* property). A PRC is *public-key* if the encoding procedure is public (whereas detection requires the corresponding secret key).

In this paper, we provide the first construction of public-key PRCs for the binary alphabet satisfying robustness in the presence of a constant fraction of substitutions ($1/6 - \varepsilon$, for arbitrary $\varepsilon > 0$) and at the same time achieving pseudorandomness against sub-exponential-time distinguishers. The pseudorandomness property relies on a new family of *distorted* McEliece assumptions that we introduce, instantiated with a class of expanded subcodes of Reed-Solomon codes, called Raw Reed-Solomon codes, for which we provide heuristic evidence of (plausible) sub-exponential hardness.

Our construction is obtained by revisiting the original blueprint by Christ and Gunn to obtain public-key PRCs based on McEliece assumptions. Along the way, we also uncover that their blueprint does not work directly with Raw Reed-Solomon codes. In particular, we show that a generating matrix of a permuted Raw Reed-Solomon code is distinguishable in polynomial time from a uniformly random generating matrix. To circumvent that difficulty, we propose to *distort the public key* by multiplication with a sparse invertible matrix of constant row Hamming weight.

Keywords: pseudorandom codes; code-based cryptography; sub-exponential security; watermarking.

1 Introduction

Generative AI is transforming society in many beneficial ways, but it also raises serious concerns by enabling the creation and spread of fake news and misinformation. *Watermarking* is a technique which embeds a signal in the generated

content, in such a way that the signal can later be recovered to distinguish machine-generated content from human outputs. Detection of the watermark requires a secret key, while generating watermarked content might either require the same secret key used for detection or a corresponding public key; in the latter case, we speak of *public-key* watermarking.

In order to be useful, watermarking should meet a few basic properties: (i) *undetectability* ensures that the output of a watermarked model is computationally indistinguishable from that of the original model (and thus, the watermarking does not degrade the quality of the model); (ii) *soundness* ensures that human-generated content is not deemed as watermarked; (iii) *robustness* ensures that the watermark cannot be removed by attackers modifying the text in some ways.

Seminal work in the area [ZEF+24] shows that no watermarking scheme can achieve robustness against *arbitrary* adversarial strategies (in a very strong sense); for this reason, the research on watermarking is focused on achieving robustness against attackers that can modify the text in *somewhat limited* ways.

1.1 Pseudorandom codes

Pseudorandom codes (PRCs), introduced by Christ and Gunn [CG24], are a natural cryptographic tool that can be leveraged in order to achieve robust watermarking of arbitrary AI-generated content in a generic way. In a nutshell, a secret-key (respectively public-key) PRC is a secret-key (resp. public-key) encryption scheme with pseudorandom ciphertexts and error correction guarantees. The pseudorandomness property translates into the undetectability of the watermarking; the error-correction property translates into the robustness of the watermarking against limited modification channels. Both properties are somewhat orthogonal, which makes the design of PRCs a difficult task, and requires the introduction of non-standard assumptions or extremal regime of parameters. Additionally, the decryption of any fixed ciphertext (independent of the secret key) should result in an error, in order to ensure the soundness property of the watermarking. Trivial constructions can achieve sublinear robustness, hence, to be meaningful, a construction has to support at least a constant fraction of substitutions.

Prior constructions of PRCs in the literature, all suffer from at least one of the following limitations:

- They are not sub-exponentially secure, in that the underlying computational assumptions needed to achieve pseudorandomness can be broken by a quasi-polynomial-time distinguishing attack.
- They only achieve robustness in the presence of adversaries that have limited knowledge (or no knowledge at all) on the secret key.

The only exception is a beautiful recent work by Christ *et al.* [CGG+26] who gave a construction of a *secret-key* PRC supporting binary alphabets and achieving a strong form of robustness known as *strongly adaptive edit-robustness*, where the latter means that the robustness adversary can perform arbitrary edits (up

to a certain bound)—in the form of substitutions, deletions and insertions—and has full knowledge of the secret key. Their construction requires a new *permuted codes conjecture* that plausibly withstands sub-exponential-time distinguishing attacks. This leaves us with the following natural question:

> *Can we construct* public-key *PRCs, over the binary alphabet, achieving pseudorandomness against* sub-exponential-time *distinguishers and* strongly adaptive *robustness?*

1.2 Our contributions

Our main result is a positive answer to the above question for the case of strongly-adaptive *substitution*-robustness, where the latter means that the robustness adversary cannot modify the length of the codeword. Our construction requires a new family of assumptions that we dub the *distorted McEliece assumptions*, as it can be interpreted as a distorted variant of the standard McEliece assumption for building public-key encryption [McE78b]. We remark that, in light of the black-box separations shown in [DMR25,GGW25], the need for new assumptions is not surprising, and it is in fact a feature shared by all previous PRC constructions with strong-enough robustness. Our family of assumptions is parameterized by an integer γ, which translates into a bound of $1/(2\gamma) - \varepsilon$, for arbitrary $\varepsilon > 0$, on the number of tolerated substitutions. While the best achievable rate of substitutions would be $1/2 - \varepsilon$, as recently shown in [FGP+25], we suggest $\gamma \geq 3$ in order to achieve plausible pseudorandomness against sub-exponential-time distinguishers. This is a safe parameter choice to avoid an existing attack [COTG15], which we discuss in more detail in the technical overview below, that targets distortion levels $\gamma \leq 2 - \eta$ for any $\eta > 0$.

As a warm-up, we formalize a blueprint yielding a simple construction of public-key PRCs based on McEliece-like assumptions. The original PRC construction by Christ and Gunn [CG24] can be seen as a special case of this blueprint using Low-Density Parity-Check (LDPC) codes;[3] as we have already discussed, this construction does not achieve pseudorandomness against sub-exponential-time distinguishers. Hence, we investigate the security of the same blueprint when using so-called Raw Reed-Solomon codes [SKS19]. This family of codes was already used by Christ *et al.* [CGG+26] as an instantiation of their permuted codes conjecture, yielding a *secret-key* PRC with sub-exponential pseudorandomness and strongly-adaptive substitution-robustness with optimal rate approaching $1/2$.

On the negative side, we show that the above family of codes is not suitable in the setting of public-key PRCs. More precisely, we show that a generating matrix of a permuted Raw Reed-Solomon code is distinguishable in polynomial time from a uniformly random generating matrix. Our attack relies on square-code dimension arguments, in the spirit of state-of-the-art attacks against McEliece encryption with other structured families of codes [FGUO+13,CGG+14,COT16].

[3] This was also observed by Christ and Gunn [CG24].

To analyze the success probability of our attack, we offer a novel interpretation of Raw Reed-Solomon codes as a juxtaposition (up to permutations) of trace codes. Then, we argue that the breakthrough results of [MT23] about the dimension of the square of a trace code, still hold for the juxtaposition of trace codes, which in turn enables a distinguisher telling apart the difference of dimension between the square of Raw Reed-Solomon codes and of random codes.

On the positive side, we propose a modified blueprint yielding a construction of binary PRCs satisfying strongly-adaptive substitution-robustness under a plausible sub-exponential hardness assumption. In particular, we replace the permutation matrix in the key generation with a sparse invertible matrix such that each row has constant Hamming weight γ. We call this matrix a *distortion matrix* of weight γ. The pseudorandomness of this new construction relies on a new assumption, the *distorted McEliece assumption*, which is weaker than the previous assumption used in the original blueprint. We conjecture that the family of Raw Reed-Solomon codes with a distortion matrix of weight $\gamma \geq 3$ is enough to circumvent known distinguishing attacks, thus yielding plausible sub-exponential hardness, which translates into sub-exponential pseudorandomness (under the corresponding distorted McEliece assumption). We provide several experimental arguments in favor of this conjecture. Turning to robustness, we show that the presence of a distortion matrix penalizes the robustness rate by a factor $1/\gamma$. (The soundness property can still be proven under the same conditions.) As a result, we achieve the first public-key PRC construction with strongly-adaptive robustness up to a constant $1/(2\gamma)$-fraction of substitutions, and pseudorandomness under a plausibly subexponentially hard assumption.

1.3 Overview of techniques

Below, we provide a more technical overview of our results.

The blueprint of [CG24] for public-key PRCs. We start, in Section 3, by revisiting the zero-bit[4] public-key PRC construction of [CG24] under the McEliece framework and abstract it for a generic family $\mathcal{F}$ of codes. The construction works as follows: the key generation samples a code $\mathcal{C}$ from $\mathcal{F}$ and applies a random secret permutation on the coordinates of the codewords of $\mathcal{C}$, resulting in a public code $\mathcal{C}' = \mathcal{C}\,\mathbf{P}^{-1}$. Encoding proceeds with sampling a word from $\mathcal{C}'$ and adding some Bernoulli noise.[5] To decode, the received word is permuted with the secret $\mathbf{P}$, then a decoding algorithm from $\mathcal{C}$ is used to detect whether this results in a valid codeword or a uniform vector. Informally, the algorithms write as follows:

$\mathsf{KeyGen}(1^\lambda)$**:** The key-generation algorithm outputs $\mathsf{pk} = \mathcal{C}' = \mathcal{C}\mathbf{P}^{-1}$ and $\mathsf{sk} = (\mathcal{C}, \mathbf{P})$ where $\mathcal{C} \hookleftarrow \mathcal{F}$ and $\mathbf{P} \hookleftarrow \mathcal{S}_n$.

[4] Zero-bit PRC means there is a single public message, and is sufficient for implementing watermarking schemes [CG24].

[5] The uniform offset vector $\mathbf{o}$, required for soundness, is omitted for clarity in this overview.

$\mathsf{Encode}(\mathsf{pk})$**:** The encoding algorithm samples $\mathbf{x} \hookleftarrow \mathcal{C}'$, $\mathbf{e} \hookleftarrow \chi$ for some Bernoulli error distribution χ and finally it outputs the codeword $\mathbf{y} = \mathbf{x} + \mathbf{e}$.

$\mathsf{Decode}(\mathsf{sk}, \mathbf{y})$**:** The decoding algorithm computes $\mathbf{y}' = \mathbf{y}\mathbf{P}$ and executes the decoding algorithm of $\mathcal{C}$ on $\mathbf{y}'$. If it succeeds, return $\top$; otherwise, return $\bot$.

The pseudorandomness of this construction is exactly equivalent to the decisional McEliece Problem McEP (Definition 10) which asks to distinguish between distributions $(\mathbf{G}, \mathbf{x}\mathbf{G}+\mathbf{e})$ and $(\mathbf{G}, \mathbf{y})$ where $\mathbf{G}$ is a generating matrix of the public code $\mathcal{C}'$ output by the key generation algorithm, $\mathbf{x}\mathbf{G}+\mathbf{e}$ is a noisy codeword from $\mathcal{C}'$ and $\mathbf{y}$ is a uniform vector. The family $\mathcal{F}$ of codes is a parameter of the system and of the problem McEP, whose hardness is therefore dependent on the actual choice of the family $\mathcal{F}$.

We also define the corresponding McEliece Public Key Problem McEPKP (Definition 12) which asks to distinguish between $\mathbf{G}$, a generating matrix of the public code, and a uniformly random matrix. A simple hybrid argument yields that the hardnesses of McEPKP and of the standard LPN problem imply the hardness of McEP; for this reason, it is common to consider the hardness of McEPKP for the pseudorandomness property of the PRC construction.

When it comes to choosing a specific family $\mathcal{F}$ of codes, the original McEliece construction [McE78a] uses the family of Goppa codes, and it is usually believed that the corresponding McEP with Goppa codes is sub-exponentially hard. However, Goppa codes achieve too low a level of robustness, because they can correct only a sub-linear fraction of substitutions. Other families of codes with more structure, such as Generalized Reed-Solomon codes or Algebraic Geometry codes, have been proposed for McEliece encryption, and would improve the robustness of PRCs, but have all been broken in polynomial or quasi-polynomial time [SS92, CMCP14]. The PRC construction of Christ and Gunn [CG24] uses the family of LDPC codes with $O(\log n)$ dual weight, which achieve much better robustness than Goppa, but whose pseudorandomness is quasi-polynomially broken.

Inadequacy of Raw Reed-Solomon codes in the blueprint of [CG24]. In Section 4, we investigate an alternative natural candidate for the family of codes $\mathcal{F}$ in the above blueprint: the family of Raw Reed-Solomon codes [SKS19], a special case of expanded codes. This family of codes has recently been proposed by Christ *et al.* [CGG+26] as an instantiation of the permuted codes conjecture, which yields secret-key PRCs with sub-exponential pseudorandomness and strongly-adaptive substitution-robustness.

Expanded codes are a construction to turn a code from a large alphabet $\mathbb{F}_{q^m}$ and length n into a code from a small alphabet $\mathbb{F}_q$ with longer length nm. An isomorphism Φ from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q^m$ is applied on each of the n symbols on the initial code, and the outputs are concatenated into a codeword of length nm. Starting with a subcode (of codimension 1) of a Reed-Solomon code over $\mathbb{F}_{q^m}$ consisting of the evaluation of polynomials belonging to the subspace of polynomials with 0 as a constant factor, then expanding it into an expanded code over $\mathbb{F}_q$ for any isomorphism, defines a Raw Reed-Solomon code. Such codes have constant-size

alphabet and are able to decode up to any $\frac{1}{2} - \varepsilon$ fraction of errors for any small constant $\varepsilon > 0$.

Unfortunately, we prove that McEPKP becomes an easy problem when instantiated with the family of Raw Reed-Solomon codes, and more generally for any family of expanded codes. We establish this result thanks to an alternative description of expanded codes with the field trace operator, which enables the adaptation of proof techniques from [MT23] to analyze the dimension of the square of trace codes. More precisely, we can think of the field trace operator for the extension field $\mathbb{F}_{q^m}/\mathbb{F}_q$ as offering an alternative viewpoint of expanded codes: as observed in [CL21], each symbol x of the initial code over $\mathbb{F}_{q^m}$ is replaced with its trace coordinates $\mathrm{Tr}(b_1^* x), \ldots, \mathrm{Tr}(b_m^* x)$ for some basis $(b_1^*, \ldots, b_m^*)$ of $\mathbb{F}_{q^m}$ as an $\mathbb{F}_q$-linear space, which depends only on the choice of the isomorphism Φ. By rearranging the indices of the coordinates, the resulting expanded codeword is a concatenation of m trace operators $(\mathrm{Tr}(b_1^* \mathbf{c}), \ldots, \mathrm{Tr}(b_m^* \mathbf{c}))$ where $\mathbf{c}$ is a codeword from $\mathcal{C}$ and the trace operator is naturally applied coordinate-wise to vectors. In [MT23], the authors start by proving some results on the dimension of the square of trace codes, and then apply it to prove an upper bound on the dimension of the square of the dual of alternant and Goppa codes of high-rate, because they are trace code of generalized Reed-Solomon codes. Expanded codes are not exactly trace codes. However, because they consist of the concatenation of various trace operators applied on codewords, we were able to obtain the same results as in [MT23] regarding the dimension of the square of expanded codes, and apply it to the case of Raw Reed-Solomon codes. We get the following theorem that is proven in Section 4.

Theorem 1 (Informal). *Let $\mathcal{C}$ be a Raw Reed-Solomon code of dimension km, expanded from a Reed-Solomon code of dimension k over the alphabet $\mathbb{F}_{q^m}$. Then, the square of $\mathcal{C}$, denoted $\mathcal{C}^{\star 2}$, has dimension* $\dim(\mathcal{C}^{\star 2}) \leq \binom{km+1}{2} - \frac{m}{2}(k-1)(k-2)$.

The above upper bound on the dimension, compared to the typical case of random codes where the dimension of the square code is exactly $\binom{km+1}{2}$, gives a polynomial-time distinguisher for McEPKP with Raw Reed-Solomon codes. We can think of this as strong evidence that the corresponding McEP is likely to be easy with this family of codes, making it unsuitable for use in either PRCs in the blueprint of [CG24], or in McEliece encryption.

Distorting the public key to prevent distinguishing attacks. To overcome the distinguisher on Raw Reed-Solomon codes, we propose to replace the permutation matrix $\mathbf{P}$ in the key generation with a sparse invertible matrix such that each row has constant integer Hamming weight γ. We call this matrix a *distortion matrix* of weight γ. This is a natural generalization of permutations, as the set of permutation matrices is exactly the set of sparse invertible matrices such that each row has Hamming weight 1. We conjecture that a distortion matrix of weight $\gamma \geq 3$ is enough to circumvent the previous distignuishing attacks on Raw Reed-Solomon codes (see below for a justification of the minimal distortion value 3). We thus define a new assumption, the *distorted*

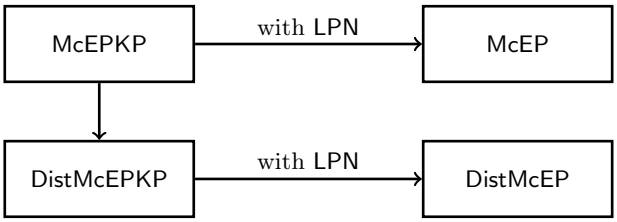


Fig. 1: Relation between McEliece assumptions and our new distorted flavours. The graphical notation $\mathsf{A} \to \mathsf{B}$ means that the hardness of assumption A implies the hardness of assumption B; similarly, $\mathsf{A} \xrightarrow{\text{with LPN}} \mathsf{B}$ has the same meaning, except that the hardness relation between A and B also assumes the hardness of LPN.

McEliece assumption, DistMcEP, which still asks to distinguish between distributions $(\mathbf{G}, \mathbf{xG} + \mathbf{e})$ and $(\mathbf{G}, \mathbf{y})$, with the difference that $\mathbf{G}$ is the generating matrix of a code $\mathcal{C}' = \mathcal{C}\,\mathbf{P}^{-1}$ that is now computed as a code from the family $\mathcal{F}$ multiplied with the inverse of a distortion matrix instead of a permutation. We also define the corresponding assumption for the public key DistMcEPKP. This latter assumption is weaker than McEPKP, hence DistMcEP is somewhat weaker than McEP. The relation between the standard and our new distorted flavour is summarized in Figure 1.

The idea of distorting a generating matrix to produce an indistinguishable McEliece public key has already been studied several times from a cryptanalysis point of view. In [BBC+16], a Generalized Reed-Solomon matrix is distorted by a matrix $(\mathbf{R} + \mathbf{T})^{-1}$, where $\mathbf{R}$ is of rank 1 and $\mathbf{T}$ is sparse of weight γ.[6] However, for efficiency reasons they were obliged to take $\gamma < 2$ which led to a polynomial attack [COTG15]. In this paper we simplify (by dropping the rank 1 matrix $\mathbf{R}$) and extend the distortion for larger $\gamma \geq 3$ values, making it resistant to [COTG15]. In rank metric, an alternative to the Hamming metric in coding theory, Loidreau's scheme [Loi17] uses a distortion matrix. Attacks to remove the distortion have been studied [CC20, Loi22, BL23] but up to this date the scheme remains secure even for small distortion values $\gamma = 3$ or $\gamma = 4$.

In Section 5.1, we provide several arguments in favor of the hardness of the distorted McEliece Problem using the family of Raw Reed-Solomon codes with distortion weight $\gamma \geq 3$. First, as already mentioned, the polynomial-time attack of [COTG15] is not feasible. Second, we give experimental evidence that the square code of a distorted Raw Reed-Solomon has the same dimension as the square of a random code.[7]

[6] In their paper, the distortion parameter γ is actually denoted m, and the distortion parameter is allowed to be decimal, e.g. 1.6.

[7] It is known to be difficult to prove results about square code dimensions in new McEliece designs, therefore we could not give a formal proof and rely on experiments.

A refined blueprint for public-key PRCs. Finally, in Section 6, we modify the blueprint of [CG24] for public-key PRCs, by replacing the permutation matrix with a distortion matrix in the key generation. This yields a refined construction for zero-bit public-key PRCs whose pseudorandomness relies on the DistMcEP. When instantiated using the family of Raw Reed-Solomon codes it enjoys strongly-adaptive substitution-robustness up to a constant $1/(2\gamma)$ fraction of substitutions, where γ is the weight of the distortion matrix. We prove the informal theorem below that summarizes all these properties.

Theorem 2 (Informal). *Assuming hardness of the Distorted McEliece Problem* DistMcEP *for the family of Raw Reed-Solomon codes and a constant distortion parameter $\gamma \geq 3$, the PRC from Algorithm 2 is pseudorandom, sound and has strongly-adaptive substitution-robustness to any channel introducing at most $1/(2\gamma) - \varepsilon$ fraction of substitutions for any small constant $\varepsilon > 0$.*

We highlight that zero-bit PRCs are sufficient for building watermarking schemes as described in [CG24]. In Section 7, we show how to support multi-bit messages with our construction.

We point out that using our new family of distorted Raw Reed-Solomon codes in the context of McEliece encryption is out of scope of this work, but would probably not be useful to improve the parameters of Classic McEliece [ABC+22]. This is because while Raw Reed-Solomon codes may have a better decoding capacity asymptotically, the actual performance might actually be worse for practical parameters due to the presence of large constant factors.

From zero-bit PRCs to multi-bit PRCs. The authors of [CG24] show how to turn a zero-bit PRC into a multi-bit PRC with constant information rate. Unfortunately, their construction does not support strong adaptive robustness but only a weaker form of non-adaptive robustness where the attacker has no access to the public key when choosing the distribution of the errors.

In [AAC+25], a multi-bit public-key construction is presented with adaptive robustness, but it crucially relies on a single-bit PRC from LDPC codes that is broken in quasi-polynomial time, as mentioned before. Moreover, the resulting multi-bit construction has somewhat low adaptive robustness tolerating only $\leq 1/32$ fraction of substitutions.

In Section 7, we show how to obtain a multi-bit PRC under the same assumptions of our zero-bit construction. Intuitively, we first turn our zero-bit PRC construction into a multi-bit PRC supporting uniformly random messages. Then, by relying on a public-key encryption (PKE) scheme with pseudorandom ciphertexts, we generically compile the latter into a multi-bit PRC supporting arbitrary messages. Our final construction comes at the cost of diving by 2 the robustness rate of the zero-bit construction, because our multi-bit PRC construction requires a unique decoder for the considered family of codes, and while there exist bounded-distance decoders that correct up to a $1/2$-fraction of errors, a unique decoder can only correct up to a $1/4$-fraction of errors. As a result, we can obtain a multi-bit public-key PRC supporting arbitrary messages with strong adaptive robustness approaching a $1/12$-fraction of errors.

1.4 Related Work

Watermarking for generative AI was initially formalized by Kirchenbauer *et al.* [KGW+23], with many follow-up constructions proposed in later works [CGZ24, ZALW24, KTHL24]. The constructions based on PRCs [CG24, GG25, AAC+25, CGG+26] were the first to obtain at the same time pseudorandomness and robustness. We refer the reader to Table 1 for a comparison between all known PRC constructions and our construction. Christ and Gunn [CG24] gave a construction of public-key PRCs with robustness approaching up to a 1/2-fraction of substitutions/deletions; the pseudorandomness property holds under the LPN assumption and the planted XOR assumption. Alternatively, as already observed by [CG24], we can think of this construction as being based on the McEliece assumption instantiated with LDPC codes.

Later, Golowich and Moitra [GM24] provided a secret-key PRC satisfying even stronger edit-robustness with rate approaching 1/2 and pseudorandomness under the assumption that there exist $(\log n)$-local weak PRFs. However, their PRC does not support the binary alphabet but rather requires a polynomial-size alphabet; the latter translates into an unrealistic assumption on the per-word entropy in the watermarking application. Ghentiyala and Guruswami [GG25] construct PRCs with optimal-rate substitution-robustness and pseudorandomness under the assumption that a random hypergraph is indistinguishable from a random hypergraph with a planted structure consisting of $\log n$ hyperedges. Alrabiah *et al.* [AAC+25] proved that the original construction from [CG24] can be transformed into a PRC satisfying robustness in the presence of a computationally-bounded adversary that can adaptively query the encoding/decoding oracles (this is weaker than strongly-adaptive robustness). Their construction actually yields a one-bit PRC with robustness rate approaching 1/4 (which is optimal for one-bit PRCs assuming the attacker has access to the encoding oracle). In this work, similar to most of the related constructions, we focus on zero-bit constructions, as they are sufficient for many practical applications, although we believe that a multi-bit construction can be easily obtained from our zero-bit constuction.

All of the above constructions suffer from the limitation that the pseudorandomness property can be broken in quasi-polynomial time. Christ *et al.* [CGG+26] were the first to obtain PRCs with plausible sub-exponential pseudorandomness under the permuted codes conjecture. They gave two constructions: one achieving rate-optimal substitution robustness, and one achieving constant-rate (albeit for a small constant) edit-robustness. Their permuted codes assumption is close to the McEliece problem as it states the hardness of distinguishing between a noisy codeword and a uniform vector, but it differs in two fundamental ways: (i) the attacker does not have access to a description of the code in the permuted code assumption; (ii) the noisy codeword is actually obtained after applying an alphabet permutation to every symbol of the codeword. This last ingredient completely destroys the linearity of the obtained samples and therefore is an obstacle in making their construction public-key. By changing the assumption, we propose a public-key construction, although we only have 1/6 robustness rate.

Reference	Public-Key?	Pseudorandomness		Robustness		
		Assumption	Sub-exp?	Type	Rate	Strongly-adaptive?
[CG24]	✓	McEP(LDPC)*	✗	Sub./Del.	$1/2 - \varepsilon$	✗
[GM24]	✗	Local weak PRFs	✗	Edit†	$1/2 - \varepsilon$	✗
[GG25]	✓	Planted Hyperloop	✗	Sub.	$1/2 - \varepsilon$	✗
[AAC+25]	✓	McEP(LDPC)*	✗	Sub.	$1/4 - \varepsilon$	✗‡
[CGG+26]	✗	Permuted Puzzles	✓	Edits	1/400	✓
	✗	Permuted Puzzles	✓	Sub.	$1/2 - \varepsilon$	✓
Section 6	✓	DistMcEP(RRS)	✓	Sub.	$1/6 - \varepsilon$	✓

Table 1: Comparing zero-bit PRCs from the literature with our construction. LDPC: Low-Density Parity Check. RRS: Raw Reed-Solomon. Sub-exp? defines whether the assumption for pseudorandomness is sub-exponentially hard or not. The robustness type specifies the type of tolerated operations: substitutions (Sub.), deletions (Del.), insertions, or edits (all of the three operations). The value $\varepsilon > 0$ is an arbitrarily small constant. *For some constructions, McEP(LDPC) actually follows from LPN (although in a regime that is broken in quasi-polynomial time). †The construction in [GM24] does not support the binary alphabet. ‡While the construction in [AAC+25] does not achieve strongly-adaptive robustness, it resists attackers that can adaptively query both the encoding and the decoding oracles.

Approaching the optimal 1/2 rate with a zero-bit public-key construction with plausible sub-exponential security is still an open problem.

Garg, Gunn and Wang [GGW25], concurrently with Döttling, Müller, and Rajasree [DMR25], established that PRCs cannot be constructed in a black-box manner from a wide range of cryptographic primitives including random oracles, public-key encryption, and virtual black-box obfuscation. In this light, it is not surprising that all prior works constructing PRCs, including ours, relied on somewhat exotic assumptions.

2 Preliminaries

2.1 Notation

Let $\mathbb{F}_q$ be the finite field of size q. Vectors are assumed to be row vectors unless stated otherwise and are denoted with lowercase bold letters and matrices with uppercase bold letters. $\star$ denotes the component-wise product of two vectors of the same length. Let $\mathcal{GL}_n(\mathbb{F}_q)$ be the set of non-singular matrices of $\mathbb{F}_q^{n\times n}$, $\mathcal{S}_n$ the set of permutation matrices of $\mathbb{F}_q^{n\times n}$, and $\mathcal{SP}_{\gamma,n}(\mathbb{F}_q)$ the set of sparse square matrices of $\mathbb{F}_q^{n\times n}$ of row weight γ (i.e. each row contains exactly γ non-zero symbols). When P is a polynomial, $\deg(P)$ denotes its degree. We let $d(\mathbf{x}, \mathbf{y})$

denote the Hamming distance between two vectors $\mathbf{x}$ and $\mathbf{y} \in \mathbb{F}_q^n$, *i.e.*,

$$d(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \sharp\{i \in [1, n] : x_i \neq y_i\}.$$

The Hamming weight $|\mathbf{x}|$ is a shorthand for $d(\mathbf{x}, \mathbf{0})$. When $S \subset \mathbb{F}_q^n$ is a set of vectors, we denote the distance between $\mathbf{x}$ and S as $d(\mathbf{x}, S) \stackrel{\text{def}}{=} \min_{\mathbf{y} \in S} d(\mathbf{x}, \mathbf{y})$. When X is a finite set, $x \hookleftarrow X$ means that x is sampled uniformly in the set X. For $\eta \in [0, 1]$, the q-Bernoulli distribution is a generalization over $\mathbb{F}_q$ of the binary Bernoulli distribution. With probability η, it samples uniformly a non-zero element of $\mathbb{F}_q$, and with probability $1 - \eta$, it outputs 0. $\mathcal{B}_\eta(\mathbb{F}_q^n)$ denotes the distribution over $\mathbb{F}_q^n$ where each coordinate is an i.i.d. q-Bernoulli random variable of parameter η. For $x \in (0, 1)$, we denote the q-ary entropy function as $H_q(x) = x \log_q(q-1) - x \log_q(x) - (1-x) \log_q(1-x)$.

2.2 Linear Codes

We start by recalling the definition of linear codes and permutation equivalence of codes. A linear code is an error-correcting code whose codewords form a vector subspace over a finite field. Equivalently, any linear combination of valid codewords is still a valid codeword, allowing encoding and decoding to be expressed using linear algebra.

Definition 1. *A linear code $\mathcal{C}$ of length n is a $\mathbb{F}_q$-subspace of $\mathbb{F}_q^n$ of dimension k. It is denoted $\mathcal{C}[n, k]_q$ as a shorthand, or even $\mathcal{C}[n, k]$ when the size of the field $\mathbb{F}_q$ is clear from the context.*

Definition 2 (Permutation equivalence). *Two linear codes $\mathcal{C}, \mathcal{C}'$ are permutation equivalent when there exists a permutation matrix $\mathbf{P} \in \mathcal{S}_n$ such that $\mathcal{C}' = \mathcal{C}\,\mathbf{P}$. The relation is denoted $\mathcal{C} \sim_\pi \mathcal{C}'$.*

Square codes. In this work, we will also make use of the notion of square codes derived from an initial linear code.

Definition 3 (Square codes). *When $\mathcal{C}[n, k]$ is a linear code, the square of $\mathcal{C}$ is defined as the code spanned by all the component-wise product between any pair of codewords in $\mathcal{C}$, i.e., $\mathcal{C}^{\star 2} \stackrel{\text{def}}{=} \{\mathbf{c} \star \mathbf{d} \,|\, \mathbf{c}, \mathbf{d} \in \mathcal{C}\}$, where $\star$ denotes the component-wise product of two vectors in $\mathbb{F}_q^n$.*

As demonstrated in [MT23], we have an upper-bound on the dimension of $\mathcal{C}^{\star 2}$.

Proposition 1 ([MT23, Prop. 10]). *When $\mathcal{C}[n, k]$ is a linear code of dimension k, $\mathcal{C}^{\star 2}$ is a linear code of the same length n and its dimension verifies*

$$\dim(\mathcal{C}^{\star 2}) \leq \min\left\{ \binom{k+1}{2}, n \right\}.$$

Moreover, the square codes derived from any two permutation equivalent linear codes are also permutation equivalent.

Proposition 2. *If two codes $\mathcal{C} \sim_\pi \mathcal{C}'$ are permutation equivalent, then their respective square codes are equivalent under the same permutation $\mathcal{C}^{\star 2} \sim_\pi \mathcal{C}'^{\star 2}$.*

Proof. This property easily follows from the fact that component-wise and permutation commute; for any $\mathbf{c}, \mathbf{d} \in \mathbb{F}_q^n$, and $\mathbf{P} \in \mathcal{S}_n$, $(\mathbf{cP}) \star (\mathbf{dP}) = (\mathbf{c} \star \mathbf{d})\mathbf{P}$. □

Some codes come with an efficient decoding algorithm which, given a vector $\mathbf{y}$, outputs another vector which lies in $\mathcal{C}$ and is close to $\mathbf{y}$ in Hamming distance. We are interested in this paper in bounded-distance decoding, where the decoding algorithm succeeds for any $\mathbf{y}$ that is close to the code. This is in contrast with weaker forms of decoding where success is only granted when $\mathbf{y}$ is produced from the sum of an actual codeword and some noise vector from a specific channel (e.g. binary symmetric channel).

Definition 4 (Bounded-distance decoding). *A code $\mathcal{C}$ has bounded-distance decoding property up to a fraction $\delta \in (0,1)$ of errors efficiently if there exists a PPT algorithm $\mathsf{Decode}_\mathcal{C} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ such that for any $\mathbf{y} \in \mathbb{F}_q^n$ with $d(\mathbf{y}, \mathcal{C}) \leq \delta n$, we have $\hat{\mathbf{y}} = \mathsf{Decode}_\mathcal{C}(\mathbf{y}) \in \mathcal{C}$ and $|\hat{\mathbf{y}} - \mathbf{y}| \leq \delta n$ with probability $\geq 1 - 2^{-\Omega(n)}$ over the randomness of $\mathsf{Decode}_\mathcal{C}$.*

One of the most popular efficient codes are Reed-Solomon codes. They are defined over a field $\mathbb{F}_{q^m}$ such that $n < q^m$ and have bounded-distance decoding up to a fraction of $\frac{1}{2}(1 - \frac{k}{n})$ of errors.

Definition 5 (Reed-Solomon Codes). *Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ be distinct elements of $\mathbb{F}_{q^m}$ with $n < q^m$. The $[n,k]_{q^m}$ Reed-Solomon code of support $\mathbf{x}$ is*

$$\mathsf{RS}_k(\mathbf{x}) \stackrel{\text{def}}{=} \{(P(x_1), \ldots, P(x_n)) \,|\, P \in \mathbb{F}_{q^m}[X], \deg(P) < k\}.$$

Expanded Codes. For practical applications we sometimes require efficient decodable codes whose field size is a constant, thus Reed-Solomon codes may not be suitable. Expanded codes are a generic transformation which turns a code with a large field size into a longer code with a smaller field size.

Definition 6 (Expanded codes). *Let $\mathbb{F}_{q^m}$ be an extension field of degree m and $\Phi : \mathbb{F}_{q^m} \to \mathbb{F}_q^m$ be a $\mathbb{F}_q$-isomorphism. Let $\mathcal{C} \subset \mathbb{F}_{q^m}^n$ be a linear code. The expanded code of $\mathcal{C}$ is a linear code of $\mathbb{F}_q^{mn}$ defined as*

$$\mathbf{Exp}_\Phi(\mathcal{C}) \stackrel{\text{def}}{=} \{(\Phi(c_1), \ldots, \Phi(c_n)) \,|\, \mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{C}\}.$$

A basis $\mathcal{B} = (b_1, \ldots, b_m)$ of $\mathbb{F}_{q^m}$ is canonically associated to a $\mathbb{F}_q$-isomorphism via coordinates $\Phi_\mathcal{B} : x = \sum_i x_i b_i \mapsto (x_1, \ldots, x_m)$. We denote $\mathbf{Exp}_\mathcal{B}(\mathcal{C})$ the expanded code using basis $\mathcal{B}$, i.e., $\mathbf{Exp}_\mathcal{B}(\mathcal{C}) \stackrel{\text{def}}{=} \mathbf{Exp}_{\Phi_\mathcal{B}}(\mathcal{C})$.

An expanded code (Definition 6) can be viewed as a concatenation of trace codes, up to reordering of coordinate indices, as shown in Proposition 3. Before proceeding with its proof, we formally define the trace operator and trace codes.

Definition 7. *The trace operator of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ defines as the $\mathbb{F}_q$-linear map over $\mathbb{F}_{q^m}$*

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(x) \stackrel{\text{def}}{=} \sum_{i=0}^{m-1} x^{q^i}.$$

We will omit the extension field $\mathbb{F}_{q^m}/\mathbb{F}_q$ and simply write $\mathrm{Tr}(\cdot)$, whenever the former is clear from the context. The definition extends naturally to vectors of $\mathbb{F}_{q^m}^n$ and to $\mathbb{F}_{q^m}$-linear codes:

$$\mathrm{Tr}(\mathbf{x}) \stackrel{\text{def}}{=} (\mathrm{Tr}(x_1), \ldots, \mathrm{Tr}(x_n)) \text{ and } \mathrm{Tr}(\mathcal{C}) \stackrel{\text{def}}{=} \{\mathrm{Tr}(\mathbf{c}) \,|\, \mathbf{c} \in \mathcal{C}\}.$$

For $\mathcal{F} = (f_1, ..., f_m)$ a family of scalars in $\mathbb{F}_{q^m}$, we define $\mathrm{Tr}_{\mathcal{F}}(\mathcal{C})$ as follows:

$$\mathrm{Tr}_{\mathcal{F}}(\mathcal{C}) \stackrel{\text{def}}{=} \{(\mathrm{Tr}(f_1\mathbf{c}), \ldots, \mathrm{Tr}(f_m\mathbf{c})) \,|\, \mathbf{c} \in \mathcal{C}\}.$$

Proposition 3. *If $\mathcal{B}^* = (b_1^*, \ldots, b_m^*)$ denotes the dual basis of $\mathcal{B}$, then we have $\mathbf{Exp}_{\mathcal{B}}(\mathcal{C}) \sim_\pi \mathrm{Tr}_{\mathcal{B}^*}(\mathcal{C})$.*

Proof. Let $\mathbf{x} \in \mathbf{Exp}_{\mathcal{B}}(\mathcal{C})$, which means there exists $\mathbf{c} \in \mathcal{C}$ such that $\mathbf{x} = \mathbf{Exp}_{\mathcal{B}}(\mathbf{c})$. As shown in [CL21],

$$\mathbf{Exp}_{\mathcal{B}}(\mathbf{c}) = (\mathrm{Tr}(b_1^*c_1), \ldots, \mathrm{Tr}(b_m^*c_1), \ldots, \mathrm{Tr}(b_1^*c_n), \ldots, \mathrm{Tr}(b_m^*c_n)).$$

We can rearrange indices to group coordinates with similar b_i^*, which means we can build a permutation $\mathbf{P}$ such that

$$\begin{aligned}\mathbf{Exp}_{\mathcal{B}}(\mathbf{c})\,\mathbf{P} &= (\mathrm{Tr}(b_1^*c_1), \ldots, \mathrm{Tr}(b_1^*c_n), \ldots, \mathrm{Tr}(b_m^*c_1), \ldots, \mathrm{Tr}(b_m^*c_n)) \\ &= (\mathrm{Tr}(b_1^*\mathbf{c}), \ldots, \mathrm{Tr}(b_m^*\mathbf{c})) \in \mathrm{Tr}_{\mathcal{B}^*}(\mathcal{C}).\end{aligned}$$

This proves that $\mathbf{Exp}_{\mathcal{B}}(\mathcal{C})\ \mathbf{P} \subset \mathrm{Tr}_{\mathcal{B}^*}(\mathcal{C})$. We can prove similarly the reverse inclusion, which finally proves the result $\mathbf{Exp}_{\mathcal{B}}(\mathcal{C}) \sim_\pi \mathrm{Tr}_{\mathcal{B}^*}(\mathcal{C})$. □

Raw Reed-Solomon Codes. Applying the expansion transformation on plain Reed-Solomon codes unfortunately does not yield the same efficient decoding properties. However, as shown in [SKS19], applying the expansion to some well-chosen subcodes of Reed-Solomon codes, makes them more powerful in terms of decoding. They are called Raw Reed-Solomon codes, and in [SKS19], two variants of Raw Reed-Solomon codes are defined. We recall the formal definition below.

Definition 8 (Raw Reed-Solomon Codes [SKS19]). *Let $\mathbf{x} = (x_1, \ldots, x_n) \in (\mathbb{F}_{q^m}^\times)^n$ be distinct non-zero elements of $\mathbb{F}_{q^m}$, $\mathcal{B} = (b_1, \ldots, b_m)$ be a basis of $\mathbb{F}_{q^m}$, and $\mathsf{SimpleRS}_k(\mathbf{x}) \subset \mathsf{RS}_{k+1}(\mathbf{x})$ be the following subcode of Reed-Solomon code $\mathsf{RS}_{k+1}(\mathbf{x})$:*

$$\mathsf{SimpleRS}_k(\mathbf{x}) \stackrel{\text{def}}{=} \{(P(x_1), \ldots, P(x_n)) \,|\, P(X) = XQ(X), Q \in \mathbb{F}_{q^m}[X], \deg(Q) < k\}.$$

The $[n, k]$ Raw Reed-Solomon code of support $\mathbf{x}$ and expansion basis $\mathcal{B}$ is defined as $\mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{Exp}_{\mathcal{B}}(\mathsf{SimpleRS}_k(\mathbf{x}))$.

Remark 1. In [SKS19], two variants of Raw Reed-Solomon codes are defined: $\mathsf{SimpleRawRS}$ and $\mathsf{OddRawRS}$. Because they both achieve the same decoding properties, we chose to focus on the simple variant in this work, but all our results would still hold when replacing $\mathsf{SimpleRawRS}$ with $\mathsf{OddRawRS}$.

We refer the reader to Appendix B.1 for the formal proof of the following statement regarding the bounded-distance decodability of Raw Reed-Solomon codes. It classically follows from their list-decodability property.

Proposition 4. *For any constants $\varepsilon > 0, \alpha \in (0,1)$, for a sufficiently large n, with $k = n^\alpha$, and for any support $\mathbf{x}$, the code $\mathcal{C} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x})$ is bounded-distance decodable from a fraction of $\frac{1}{2} - \varepsilon$ errors in the sense of Definition 4.*

2.3 Pseudorandom codes

Secret-key Pseudorandom codes (PRC) are an extension of error-correcting codes introduced by [CG24]. In a nutshell, PRCs are keyed error-correcting codes (i.e., encoding and decoding require a secret key sk output by KeyGen), which additionally guarantee the following two properties: (i) any codeword $\mathbf{c}$ independent of the secret key sk is correctly identified as invalid (soundness); i.e., decoding $\mathbf{c}$ using sk causes the decoding algorithm to output $\perp$; (ii) the keyed encoding algorithm is probabilistic and, moreover, outputs codewords that are pseudorandom in the view of a probabilistic polynomial-time adversary that does not know the secret key sk.[8]

Following the nomenclature of [CG24], in this work we focus on the *zero-bit version* of PRCs, which permits encoding a single fixed message (or, equivalently, the zero bit), i.e., the encoding algorithm does not take as input a message. Such a zero-bit notion is sufficient for building watermarking schemes as demonstrated in [CG24]. In addition, we aim to construct public-key PRCs, i.e., where encoding is public-key, while decoding remains secret-key algorithm. We refer the reader to Section 1.4 and Table 1 for the hardness assumptions and security settings considered by state-of-the-art PRC constructions.

Below, we provide the formal definition of zero-bit public-key PRCs.

Definition 9 (Zero-bit public-key PRCs). *Let Σ be an alphabet and $\mathcal{E} : \Sigma^* \times \{0,1\}^* \to \Sigma^*$ be an error channel that may take auxiliary information $\mathsf{aux} \in \{0,1\}^*$. A (zero-bit) public-key PRC with strong adaptive robustness to $\mathcal{E}$ is described by a triple of efficient randomized algorithms $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$, $\mathsf{Encode}(\mathsf{pk}) \to \Sigma^*$, $\mathsf{Decode}(\mathsf{sk}, \mathbf{y} \in \Sigma^*) \to \{\top, \perp\}$, satisfying the following properties:*

- *(Computational Pseudorandomness) For any probabilistic polynomial-time adversary $\mathcal{A}$, we have:*

$$\left| \mathbb{P}[\mathcal{A}^{\mathsf{Encode}(\mathsf{pk})}(1^\lambda, \mathsf{pk}) = 1] - \mathbb{P}[\mathcal{A}^{\mathcal{U}}(1^\lambda, \mathsf{pk}) = 1] \right| \leq \mathsf{negl}(\lambda),$$

[8] Providing sk to the adversary breaks pseudorandomness, since the adversary can distinguish a codeword from a random string by simply running the decoding algorithm.

Algorithm 1 Public-key PRC from McEliece assumptions.

Parameters: A prime power q, a family $\mathcal{F}$ of linear $[n, k]$ codes in $\mathbb{F}_q$, a noise parameter $\eta \in (0, 1)$, a correction parameter $\delta \in (0, \frac{1}{2})$.

Key-generation algorithm $\mathsf{KeyGen}(1^\lambda)$:

1. Sample a generating matrix $\mathbf{G} \hookleftarrow \mathcal{F}$ which defines a linear code $\mathcal{C}$ with decoding algorithm $\mathsf{Decode}_{\mathcal{C}}$.
2. Sample a change of basis matrix $\mathbf{S} \hookleftarrow \mathcal{GL}_k(\mathbb{F}_q)$.
3. Sample a permutation matrix $\mathbf{P} \hookleftarrow \mathcal{S}_n$.
4. Let $\mathbf{G}' = \mathbf{SGP}^{-1}$.
5. Sample $\mathbf{o} \hookleftarrow \mathbb{F}_q^n$.
6. Output $\mathsf{pk} = (\mathbf{G}', \mathbf{o})$ and $\mathsf{sk} = (\mathbf{P}, \mathbf{G}', \mathbf{o})$.

Encoding algorithm $\mathsf{Encode}(\mathsf{pk})$:

1. Let $\mathsf{pk} = (\mathbf{G}', \mathbf{o})$.
2. Sample $\mathbf{x} \hookleftarrow \mathbb{F}_q^k, \mathbf{e} \hookleftarrow \mathcal{B}_\eta(\mathbb{F}_q^n)$.
3. Output the codeword $\mathbf{y} = \mathbf{xG}' + \mathbf{e} + \mathbf{o}$.

Decoding algorithm $\mathsf{Decode}(\mathsf{sk}, \mathbf{y})$:

1. Let $\mathsf{sk} = (\mathbf{P}, \mathbf{G}', \mathbf{o})$.
2. Compute $\hat{\mathbf{y}} = \mathsf{Decode}_{\mathcal{C}}(\mathbf{y}')$ where $\mathbf{y}' = (\mathbf{y} - \mathbf{o})\mathbf{P}$.
3. If $\hat{\mathbf{y}} \in \mathcal{C}$ and $|\mathbf{y}' - \hat{\mathbf{y}}| \leq \delta n$, **return** $\top$.
4. Otherwise, **return** $\bot$.

where the probabilities are taken over the random coins of $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and* $\mathcal{A}$*. The notation* $\mathcal{A}^{\mathcal{U}}$ *means that* $\mathcal{A}$ *has access to an oracle that responds to each query with a freshly uniform random value in* Σ^**.*

– *(Soundness) For any fixed* $\mathbf{c} \in \Sigma^*$*, we have:*

$$\mathbb{P}_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)}[\mathsf{Decode}(\mathsf{sk}, \mathbf{c}) = \bot] \geq 1 - \mathsf{negl}(\lambda).$$

– *(Strong Adaptive Robustness) Let* $\mathcal{E} : \Sigma^* \times \{0, 1\}^* \to \Sigma^*$ *be the error channel defined above. Then, the following probability must hold:*

$$\mathbb{P}_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)}[\mathsf{Decode}(\mathsf{sk}, \mathcal{E}(\mathsf{Encode}(\mathsf{pk}), (\mathsf{sk}, \mathsf{pk}))) = \top] \geq 1 - \mathsf{negl}(\lambda).$$

We name the above robustness definition *strongly adaptive* due to the strong adversarial setting we consider. In particular, the error channel $\mathcal{E}$ (i.e., the adversary in the robustness experiment) can adaptively introduce errors depending on the public and secret keys that are sampled (thus, both encoding and decoding strategies are known to $\mathcal{E}$). We note that this stronger notion has been considered by [CGG+26], but only in the secret-key setting (where both encoding and decoding are secret-key). We are the first to extend and consider this stronger adversarial setting to public-key PRCs.

3 The Blueprint of [CG24] for Public-key PRCs

We revisit the construction of a zero-bit public-key PRC (Algorithm 1) by Christ and Gunn [CG24]. The decoding algorithm we consider is generic whereas in [CG24] it is specific to the LDPC family [Gal62].

In a nutshell, the scheme works as follows: (*i*) the key generation algorithm KeyGen is exactly the same as the one in McEliece public-key encryption scheme; (*ii*) the encoding algorithm Encode simply consists in sampling a McEliece ciphertext for some random message $\mathbf{x}$, while adding some public offset $\mathbf{o}$ necessary to achieve soundness; (*iii*) the decoding algorithm Decode removes the offset $\mathbf{o}$ and tries to decrypt the remaining McEliece ciphertext, returning $\top$ in case of success. The formal construction is described in Algorithm 1.

For the rest of the paper, all our constructions produce codewords of length $n = \mathsf{poly}(\lambda)$.

McEliece assumptions. The pseudorandomness of this PRC construction (Algorithm 1) relies on a computational assumption called the McEliece Problem (McEP), which is a variant of the well-known Learning Parity with Noise (LPN) problem, that asks, given a uniformly random generating matrix $\mathbf{G}$ of a $[n, k]$ linear code, to distinguish between noisy codewords or uniformly random vectors of length n.

The McEP also asks to distinguish between noisy codewords or uniformly random vectors of length n, with the major difference that the given matrix $\mathbf{G}$ is not uniformly random, but rather sampled according to a special distribution parametrized by a family $\mathcal{F}$ of codes: a code is randomly drawn from $\mathcal{F}$, then the indices of codewords are shuffled by a random permutation in $\mathcal{S}_n$, and a random generating matrix of the resulting code is output. This distribution of matrices $\mathbf{G}$ corresponds exactly to the distribution of public keys in a McEliece encryption scheme instantiated with the family $\mathcal{F}$. We formalize McEP below.

Definition 10 (Decisional McEliece Problem). *Let n, k be positive integers, q a prime power, $\eta \in (0, 1)$, and $\mathcal{F}$ be a family of matrices of $\mathcal{F} \subset \mathbb{F}_q^{k \times n}$ generating efficiently decodable linear codes in $\mathbb{F}_q^n$ of dimension k. Also, let $\mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F}}$ be a distribution that returns samples generated as follows:*

1. *Sample a generating matrix $\mathbf{G} \hookleftarrow \mathcal{F}$.*
2. *Sample a change of basis matrix $\mathbf{S} \hookleftarrow \mathcal{GL}_k(\mathbb{F}_q)$.*
3. *Sample a permutation matrix $\mathbf{P} \hookleftarrow \mathcal{S}_n$.*
4. *Output $\mathbf{G}' = \mathbf{SGP}^{-1}$.*

The Decisional McEliece Problem $\mathsf{McEP}_{q,n,k,\mathcal{F},\eta}$ asks to distinguish between distributions $(\mathbf{G}', \mathbf{xG}' + \mathbf{e})$ and $(\mathbf{G}', \mathbf{y})$, where $\mathbf{G}' \hookleftarrow \mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F}}, \mathbf{x} \hookleftarrow \mathbb{F}_q^k, \mathbf{e} \hookleftarrow \mathcal{B}_\eta(\mathbb{F}_q^n), \mathbf{y} \hookleftarrow \mathbb{F}_q^n$.

In the following, we restate the theorem of [CG24] relative to the properties of the PRC construction from Algorithm 1, in a generic way for a family $\mathcal{F}$. We provide the formal proof in Appendix B.2 since it is similar to [CG24].

Theorem 3. *Let n, k be positive integers, q a prime power, $\eta \in (0,1)$, and $\mathcal{F} \subset \mathbb{F}_q^{k\times n}$ a family of codes that can bounded-distance correct $\delta \in (0, \frac{1}{2})$ errors efficiently, such that $n(1-H_q(\delta))-k = \Omega(n)$. Assuming hardness of the McEliece Problem $\mathsf{McEP}_{q,n,k,\mathcal{F},\eta}$ for the family $\mathcal{F}$, the PRC from Algorithm 1 is pseudorandom, sound and has strong adaptive robustness to any channel introducing at most $\delta' = \delta - 2\eta$ fraction of substitutions.*

Remark 2. When δ is a constant, choosing $k = o(n)$ is sufficient to ensure the condition $n(1 - H_q(\delta)) - k = \Omega(n)$.

Introducting McEliece Public Key and LPN assumptions. It is common to consider another problem when working with McEP, namely the *McEliece Public Key problem* (McEPKP). The latter states that the public key (sampled from $\mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F}}$ as defined in Definition 10) is computationally indistinguishable from a uniformly random matrix of the same size. It is known that McEPKP, combined with LPN, implies the hardness of McEP.

Below, we provide the formal definition of LPN, McEPKP, as well as the formal statement relating both to McEP.

Definition 11 (LPN Problem). *Let n, k be positive integers, q a prime power, and $\eta \in (0,1)$. The Learning Parity with Noise Problem $\mathsf{LPN}_{q,n,k,\eta}$ asks to distinguish between distributions $(\mathbf{G}, \mathbf{xG} + \mathbf{e})$ and $(\mathbf{G}, \mathbf{y})$, where $\mathbf{G} \hookleftarrow \mathbb{F}_q^{k\times n}, \mathbf{x} \hookleftarrow \mathbb{F}_q^k, \mathbf{e} \hookleftarrow \mathcal{B}_\eta(\mathbb{F}_q^n), \mathbf{y} \hookleftarrow \mathbb{F}_q^n$.*

Definition 12 (Decisional McEliece Public Key Problem). *Let n, k be positive integers, q a prime power, and $\mathcal{F}$ be a family of matrices of $\mathcal{F} \subset \mathbb{F}_q^{k\times n}$ generating efficiently decodable linear codes in $\mathbb{F}_q^n$ of dimension k. The Decisional McEliece Public Key Problem $\mathsf{McEPKP}_{q,n,k,\mathcal{F}}$ asks to distinguish between distribution $\mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F}}$ (as defined in Definition 10) and the uniform distribution over $\mathbb{F}_q^{k\times n}$.*

Proposition 5. *If both $\mathsf{McEPKP}_{q,n,k,\mathcal{F}}$ and $\mathsf{LPN}_{q,n,k,\eta}$ are computationally hard, then $\mathsf{McEP}_{q,n,k,\mathcal{F},\eta}$ is computationally hard.*

Proof. We proceed with a series of two games. The first game is the $\mathsf{McEP}_{q,n,k,\mathcal{F},\eta}$ game and the second game is the $\mathsf{LPN}_{q,n,k,\eta}$ game. Transition between the two games involves replacing a matrix $\mathbf{G}' \hookleftarrow \mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F}}$ with uniform $\mathbf{G} \hookleftarrow \mathbb{F}_q^{k\times n}$, hence the two games are indistinguishable under hardness of $\mathsf{McEPKP}_{q,n,k,\mathcal{F}}$. □

Remark 3 (On the relation between Algorithm 1 and [CG24]). Our construction and problems McEP and McEPKP are a generalization and reconcile nicely with the results of [CG24]. Indeed, when instantiating the family $\mathcal{F}$ with LDPC codes of dual Hamming weight $t = \Theta(\log n)$, the $\mathsf{McEPKP}_{q,n,k,\mathcal{F}}$ becomes exactly the planted t-XOR assumption with r relations that was used in the first construction of PRC from [CG24]. Indeed, the planted t-XOR assumption asks to distinguish a uniformly random code from another random code, of the same dimension, whose dual contains r codewords of relative Hamming weight t. The permutation

matrix $\mathbf{P}$ used in our McEPKP only acts as a permutation on the coordinates of the planted XOR relations, leaving their Hamming weight invariant. Hence, the permutation $\mathbf{P}$ does not change the exact correspondence between McEPKP and the planted XOR assumption.

McEP with $\mathcal{F}$ equal to LDPC codes of dual Hamming weight $\Theta(\log n)$ is broken by a quasi-polynomial attack: a brute force search on all words in $\mathbb{F}_q^n$ of weight $\log n$ takes only $\binom{n}{\log n} = 2^{O(\log^2 n)}$ complexity. On the other hand, as mentioned earlier, McEP with $\mathcal{F}$ equal to Goppa codes is widely considered by the McEliece community to be subexponentially hard, but does not provide enough robustness. It is therefore natural to explore new families of codes to instantiate McEP.

4 Insecurity of the McEliece Assumption with Raw Reed-Solomon Codes

The family of Raw Reed-Solomon Codes may seem attractive for instantiating a McEliece scheme because they are binary codes with large dual minimum weights. Moreover, thanks to their high error-correction capability, they could also be used to build a PRC with robustness against a constant fraction of substitutions.

Unfortunately, in this section, we provide evidence that Raw Reed-Solomon Codes are likely insecure in a McEliece setting. In particular, we show that there exists a polynomial-time attack against McEPKP (i.e., distinguishing public keys from uniformly random matrices) when instantiating the underlying code of McEPKP with Raw Reed-Solomon Codes.

Theorem 4. *There is a polynomial-time attack against the McEliece Public Key Problem* $\mathsf{McEPKP}_{q,mn,mk,\mathcal{F}}$ *when* $\mathcal{F} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x})$ *and* $\binom{km+1}{2} < nm$.

The natural condition $\binom{km+1}{2} < nm$ ensures square codes do not fill up the whole ambient space $\mathbb{F}_q^{nm}$. Codes with small rates, which are actually required to reach optimal robustness, will meet this condition. We provide the formal proof of Theorem 4 in Section 4.1. Our polynomial distinguisher relies on computing the dimension of the square of Raw Reed-Solomon codes that is lower than in the case of random codes.

We highlight that turning our distinguisher into a key-recovery attack (thus, breaking McEP) is out of scope of this work, but could potentially be feasible thanks to the connection with alternant codes, and using the recent breakthrough techniques of [BMT23] (we see this as a potential future work). In any case, the existence of this distinguisher strongly undermines confidence in a scheme that uses McEP with Raw Reed-Solomon codes as a security assumption.

4.1 Proof of Theorem 4

As mentioned in the previous section, we prove Theorem 4 by constructing a polynomial-time distinguisher against $\mathsf{McEPKP}_{q,mn,mk,\mathcal{F}}$ that leverages the difference between the dimensions of Raw Reed-Solomon Codes and random codes.

In particular, our distinguisher relies on Theorem 5 (which we prove later in the next Section 4.2) and Theorem 6 (proved in [CCMZ15]).

Theorem 5. *Let* $\mathcal{C} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x})$. *Then, we have* $\dim(\mathcal{C}^{\star 2}) \leq \binom{km+1}{2} - \frac{m}{2}(k-1)(k-2)$.

Theorem 6 (adapted from [CCMZ15, Thm 2.3]). *Let* $\mathcal{C}$ *a random* $[mn, mk]$ *linear code with* $\binom{km+1}{2} < nm$. *Then, there exists a constant* $\delta > 0$ *such that* $\mathbb{P}\left[\dim(\mathcal{C}^{\star 2}) = \binom{km+1}{2}\right] \geq \delta$.

Given the above Theorems 5 and 6, it is easy to see the existence of a polynomial-time distinguisher against $\mathsf{McEPKP}_{q,mn,mk,\mathcal{F}}$. Specifically, upon receiving an instance $\mathbf{G}$ of $\mathsf{McEPKP}_{q,mn,mk,\mathcal{F}}$ defining a code $\mathcal{C}$ that is either a permuted Raw Reed-Solomon code or a random code, the distinguisher computes $\dim(\mathcal{C}^{\star 2})$ (note that this operation can be performed in polynomial time by computing a basis of $\mathcal{C}^{\star 2}$ via Gaussian elimination). Moreover, since the dimension of $\mathcal{C}^{\star 2}$ is invariant under permutation, if $\dim(\mathcal{C}^{\star 2}) < \binom{km+1}{2}$, the distinguisher outputs 0 (permuted Raw Reed-Solomon code); otherwise, if $\dim(\mathcal{C}^{\star 2}) = \binom{km+1}{2}$, the distinguisher outputs 1 (random code). Then, by combining Theorems 5 and 6, the advantage of the distinguisher is at least δ (where δ is a positive constant). Having said this, the next section will focus on proving Theorem 5.

4.2 Proof of Theorem 5

An upper-bound on the dimension of square of expanded codes. We start by demonstrating an upper-bound on the dimension of the square of expanded codes. This result is generic in the sense we consider any expanded codes. We state the formal result below.

Proposition 6. *Let* $\mathcal{C}$ *be a linear code over* $\mathbb{F}_{q^m}$ *and* $\mathcal{B}$ *a basis of* $\mathbb{F}_{q^m}$, *then* $\dim\left((\mathbf{Exp}_{\mathcal{B}}(\mathcal{C}))^{\star 2}\right) \leq m\dim(\mathcal{C}^{\star 2}) + \binom{m}{2}(\dim(\mathcal{C}))^2$.

This proposition is reminiscent of the formulas obtained in [MT23] regarding the square of trace codes. Thanks to the connection between expanded codes and the trace operator, which we highlighted in Proposition 3, we could obtain the same results. However, since we do not have a direct translation between expanded codes and trace codes (only the operator itself can be used), we cannot apply directly the results of [MT23]. This is essentially the reason why we demonstrate the result directly.

The proof outline for Proposition 6 will follow the blueprint below:

- We first prove Lemma 1, which states that for any two codes $\mathcal{C}, \mathcal{D}$, the Schur product of $\mathrm{Tr}_{\mathcal{F}}(\mathcal{C})$ with $\mathrm{Tr}_{\mathcal{F}}(\mathcal{D})$ is included in a sum of m linear codes.
- We then prove Lemma 2, which states that when $\mathcal{C} = \mathcal{D}$, there are symmetries in the terms appearing in the sum of Lemma 1.
- By combining Lemmas 1 and 2, we group together equal terms in the sum, and consequently obtain Lemma 3, which shows that the square of $\mathrm{Tr}_{\mathcal{F}}(\mathcal{C})$ is included in a smaller sum of $\lfloor m/2 \rfloor$ linear codes.

– In Lemma 4, we demonstrate that when m is even, the last term of the sum $\mathrm{Tr}_{\mathcal{F}}(\mathcal{C} \star \mathcal{C}^{q^{m/2}})$ is actually of smaller dimension (by a factor 2) than the expected dimension $m(\dim(\mathcal{C}))^2$.

Finally, the latter two lemmas will permit us to prove Proposition 6.

Lemma 1. *Let $\mathcal{F} = (f_1, ..., f_m)$ be a family of m scalars in $\mathbb{F}_{q^m}$ and $\mathcal{C}, \mathcal{D}$ two linear codes over $\mathbb{F}_{q^m}$ with the same length n, then $\mathrm{Tr}_{\mathcal{F}}(\mathcal{C}) \star \mathrm{Tr}_{\mathcal{F}}(\mathcal{D}) \subseteq \sum_{i=0}^{m-1} \mathrm{Tr}_{\mathcal{F}^{1+q^i}}(\mathcal{C} \star \mathcal{D}^{q^i})$ where $\mathcal{D}^{q^i} \stackrel{\text{def}}{=} \{\mathbf{d}^{q^i} \mid \mathbf{d} \in \mathcal{D}\}$ and $\mathcal{F}^u \stackrel{\text{def}}{=} (f_1^u, \ldots, f_m^u)$.*

Proof. By leveraging [MT23, Prop. 13], we have that for all $x, y \in \mathbb{F}_{q^m}$, $\mathrm{Tr}(x) \cdot \mathrm{Tr}(y) = \sum_{i=0}^{m-1} \mathrm{Tr}(x \cdot y^{q^i})$. Hence, because the star product acts component-wise, we can conclude that:

$$
\begin{aligned}
\mathrm{Tr}_{\mathcal{F}}(\mathcal{C}) \star \mathrm{Tr}_{\mathcal{F}}(\mathcal{D}) &= \{(\mathrm{Tr}(f_1\mathbf{c}) \star \mathrm{Tr}(f_1\mathbf{d}), \ldots, \mathrm{Tr}(f_m\mathbf{c}) \star \mathrm{Tr}(f_m\mathbf{d})) \mid \mathbf{c} \in \mathcal{C}, \mathbf{d} \in \mathcal{D}\} \\
&= \left\{ \sum_{i=0}^{m-1} (\mathrm{Tr}(f_1\mathbf{c} \star (f_1\mathbf{d})^{q^i}), \ldots, \mathrm{Tr}(f_m\mathbf{c} \star (f_m\mathbf{d})^{q^i})) \mid \mathbf{c} \in \mathcal{C}, \mathbf{d} \in \mathcal{D} \right\} \\
&= \left\{ \sum_{i=0}^{m-1} (\mathrm{Tr}(f_1^{1+q^i}\mathbf{c} \star \mathbf{d}^{q^i}), \ldots, \mathrm{Tr}(f_m^{1+q^i}\mathbf{c} \star \mathbf{d}^{q^i})) \mid \mathbf{c} \in \mathcal{C}, \mathbf{d} \in \mathcal{D} \right\} \\
&\subseteq \sum_{i=0}^{m-1} \left\{ (\mathrm{Tr}(f_1^{1+q^i}\mathbf{c} \star \mathbf{d}^{q^i}), \ldots, \mathrm{Tr}(f_m^{1+q^i}\mathbf{c} \star \mathbf{d}^{q^i})) \mid \mathbf{c} \in \mathcal{C}, \mathbf{d} \in \mathcal{D} \right\} \\
&= \sum_{i=0}^{m-1} \left\{ (\mathrm{Tr}(f_1^{1+q^i}\mathbf{x}), \ldots, \mathrm{Tr}(f_m^{1+q^i}\mathbf{x})) \mid \mathbf{x} \in \mathcal{C} \star \mathcal{D}^{q^i} \right\} \\
&= \sum_{i=0}^{m-1} \mathrm{Tr}_{\mathcal{F}^{1+q^i}}(\mathcal{C} \star \mathcal{D}^{q^i}).
\end{aligned}
$$

□

Lemma 2. *For any $0 \leq u < m$, and any linear code $\mathcal{C}$ over $\mathbb{F}_{q^m}$, $\mathrm{Tr}_{\mathcal{F}^{1+q^u}}(\mathcal{C} \star \mathcal{C}^{q^u}) = \mathrm{Tr}_{\mathcal{F}^{1+q^{m-u}}}(\mathcal{C} \star \mathcal{C}^{q^{m-u}})$.*

Proof. Let $\mathbf{x} \in \mathrm{Tr}_{\mathcal{F}^{1+q^u}}(\mathcal{C} \star \mathcal{C}^{q^u})$. There exist $\mathbf{c}, \mathbf{d} \in \mathcal{C}$, such that

$$\mathbf{x} = (\mathrm{Tr}(f_1^{1+q^u}\mathbf{c} \star \mathbf{d}^{q^u}), \ldots, \mathrm{Tr}(f_m^{1+q^u}\mathbf{c} \star \mathbf{d}^{q^u})).$$

Since the trace acts component-wise on vector and that $\mathrm{Tr}(x) = \mathrm{Tr}(x^{q^{m-u}})$ for any $x \in \mathbb{F}_{q^m}$,

$$\mathbf{x} = (\mathrm{Tr}(f_1^{(1+q^u)q^{m-u}}\mathbf{c}^{q^{m-u}} \star (\mathbf{d}^{q^u})^{q^{m-u}}), \ldots, \mathrm{Tr}(f_m^{(1+q^u)q^{m-u}}\mathbf{c}^{q^{m-u}} \star (\mathbf{d}^{q^u})^{q^{m-u}})).$$

By leveraging the fact that $x^{q^m} = x$ for any $x \in \mathbb{F}_{q^m}$, we obtain:

$$\mathbf{x} = (\mathrm{Tr}(f_1^{1+q^{m-u}}\mathbf{d} \star \mathbf{c}^{q^{m-u}}), \ldots, \mathrm{Tr}(f_m^{1+q^{m-u}}\mathbf{d} \star \mathbf{c}^{q^{m-u}})) \in \mathrm{Tr}_{\mathcal{F}^{1+q^{m-u}}}(\mathcal{C} \star \mathcal{C}^{q^{m-u}}).$$

Hence, $\mathrm{Tr}_{\mathcal{F}^{1+q^u}}(\mathcal{C} \star \mathcal{C}^{q^u}) \subseteq \mathrm{Tr}_{\mathcal{F}^{1+q^{m-u}}}(\mathcal{C} \star \mathcal{C}^{q^{m-u}})$ and the reverse inclusion is easily obtained. □

Lemma 3. *Let $\mathcal{F} = (f_1, ..., f_m)$ be a family of m scalars in $\mathbb{F}_{q^m}$ and $\mathcal{C}$ a linear code over $\mathbb{F}_{q^m}$, then $(\mathrm{Tr}_{\mathcal{F}}(\mathcal{C}))^{\star 2} \subseteq \sum_{i=0}^{\lfloor m/2 \rfloor} \mathrm{Tr}_{\mathcal{F}^{1+q^i}}(\mathcal{C} \star \mathcal{C}^{q^i})$.*

Proof. By leveraging Lemmas 1 and 2, we obtain the following relation:

$$\begin{aligned}(\mathrm{Tr}_{\mathcal{F}}(\mathcal{C}))^{\star 2} &\subseteq \sum_{i=0}^{m-1} \mathrm{Tr}_{\mathcal{F}^{1+q^i}}(\mathcal{C} \star \mathcal{C}^{q^i}) \text{ (by Lemma 1)} \\ &= \sum_{i=0}^{\lfloor m/2 \rfloor} \mathrm{Tr}_{\mathcal{F}^{1+q^i}}(\mathcal{C} \star \mathcal{C}^{q^i}) \text{ (by Lemma 2)}\end{aligned}$$

□

Lemma 4. *Let $\mathcal{F} = (f_1, \ldots, f_m)$ be a family of m scalars in $\mathbb{F}_{q^m}$ and $\mathcal{C}$ a linear code over $\mathbb{F}_{q^m}$, then $\dim\left(\mathrm{Tr}_{\mathcal{F}^{1+q^{m/2}}}(\mathcal{C} \star \mathcal{C}^{q^{m/2}})\right) \leq m\frac{(\dim(\mathcal{C}))^2}{2}$ if m is even.*

Proof. We will follow exactly the same proof technique as for [MT23, Prop. 16, Eq. (8)], except that we need to be especially careful about multiplications by f_i in the trace operator. Before that, we need to observe that for any $x \in \mathbb{F}_{q^m}$,

$$\left(x^{1+q^{m/2}}\right)^{q^{m/2}} = x^{1+q^{m/2}}, \tag{1}$$

because $x^{q^m} = x$. Now, let $\mathbf{c}_1, \ldots, \mathbf{c}_k$ be a basis of $\mathcal{C}$. $\mathrm{Tr}_{\mathcal{F}^{1+q^{m/2}}}(\mathcal{C} \star \mathcal{C}^{q^{m/2}})$ is generated by the $\left(\mathrm{Tr}\left(f_1^{1+q^{m/2}}\beta^{q^\ell}\mathbf{c}_i \star \mathbf{c}_j^{q^{m/2}}\right), \ldots, \mathrm{Tr}\left(f_m^{1+q^{m/2}}\beta^{q^\ell}\mathbf{c}_i \star \mathbf{c}_j^{q^{m/2}}\right)\right)$ where $(\beta, \beta^q \ldots, \beta^{q^{m-1}})$ is a normal basis of $\mathbb{F}_{q^m}$ and ℓ ranges over $[0, m-1]$ and i, j over $[1, k]$. Since for any $\alpha \in [1, m], \ell \in [0, m/2-1], i, j \in [1, k]$,

$$\begin{aligned}\mathrm{Tr}\left(f_\alpha^{1+q^{m/2}}\beta^{q^\ell}\mathbf{c}_i \star \mathbf{c}_j^{q^{m/2}}\right) &= \mathrm{Tr}\left(\left(f_\alpha^{1+q^{m/2}}\right)^{q^{m/2}}\beta^{q^{\ell+\frac{m}{2}}}\mathbf{c}_i^{q^{m/2}} \star \mathbf{c}_j\right) \\ &= \mathrm{Tr}\left(f_\alpha^{1+q^{m/2}}\beta^{q^{\ell+\frac{m}{2}}}\mathbf{c}_j \star \mathbf{c}_i^{q^{m/2}}\right) \text{ using Equation (1)}\end{aligned}$$

this implies that $\mathrm{Tr}_{\mathcal{F}^{1+q^{m/2}}}(\mathcal{C} \star \mathcal{C}^{q^{m/2}})$ is generated by the (smaller) set $\mathrm{Tr}\left(f_\alpha^{1+q^{m/2}}\beta^{q^\ell}\mathbf{c}_i \star \mathbf{c}_j^{q^{m/2}}\right)$ where ℓ ranges over $[0, m-1]$ and i, j over $[1, k]$. This set is of cardinality $m\frac{(\dim(\mathcal{C}))^2}{2}$. □

Finally, by combining Lemmas 3 and 4, we are able to demonstrate Proposition 6 (our initial proposition on the upper-bound on the dimension of square of expanded codes).

Proof of Proposition 6. $\mathbf{Exp}_{\mathcal{B}}(\mathcal{C})$ is permutation-equivalent to $\mathrm{Tr}_{\mathcal{B}^*}(\mathcal{C})$. Hence, their squares are of the same dimension $\dim\left((\mathbf{Exp}_{\mathcal{B}}(\mathcal{C}))^{\star 2}\right) = \dim\left((\mathrm{Tr}_{\mathcal{B}^*}(\mathcal{C}))^{\star 2}\right)$. By using Lemma 3, we obtain the following inequality:

$$\dim\left((\mathrm{Tr}_{\mathcal{B}^*}(\mathcal{C}))^{\star 2}\right) \leq \sum_{i=0}^{\lfloor m/2 \rfloor} \dim\left(\mathrm{Tr}_{\mathcal{B}^{*1+q^i}}(\mathcal{C} \star \mathcal{C}^{q^i})\right).$$

Following the same calculations as in [MT23, Cor. 17], and using Lemma 4 in the case of an even m, we get $\dim\left((\mathbf{Exp}_{\mathcal{B}}(\mathcal{C}))^{\star 2}\right) \leq m\dim(\mathcal{C}^{\star 2}) + \binom{m}{2}(\dim(\mathcal{C}))^2$. This concludes the proof of Proposition 6. □

An upper-bound on the dimension of square of Raw Reed-Solomon Codes. By leveraging Proposition 6 and specializing it to the case where $\mathcal{C}$ is a Raw Reed-Solomon code, we can finally demonstrate Theorem 5, which in turn implies the existence of a polynomial-time distinguisher against $\mathsf{McEPKP}_{q,mn,mk,\mathcal{F}}$ (Theorem 4) as discussed at the beginning of Section 4.

Proof of Theorem 5. Let $\mathcal{D} = \mathsf{SimpleRS}_k(\mathbf{x})$ and $\mathcal{C} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x}) = \mathbf{Exp}_{\mathcal{B}}(\mathcal{D})$. Then, we have

$$\mathcal{D}^{\star 2} \subset \{(P(x_1), \ldots, P(x_n)) \,|\, P(X) = X^2 Q(X), Q \in \mathbb{F}_{q^m}[X], \deg(Q) < 2k-1\}.$$

This implies that $\dim(\mathcal{D}^{\star 2}) \leq 2k-1$ and we can now use Proposition 6 to conclude that:

$$\begin{aligned}\dim(\mathcal{C}^{\star 2}) = \dim\left((\mathbf{Exp}_{\mathcal{B}}(\mathcal{D}))^{\star 2}\right) &\leq m\dim(\mathcal{D}^{\star 2}) + \binom{m}{2}(\dim(\mathcal{D}))^2 \\ &\leq m(2k-1) + \binom{m}{2}k^2 = \binom{km+1}{2} - \frac{m}{2}(k-1)(k-2).\end{aligned}$$

This concludes the proof for the case SimpleRawRS. □

5 The Distorted McEliece Assumption

Theorem 4 constitutes a major obstacle to building PRCs using raw Reed–Solomon codes. Proving the pseudorandomness of PRCs requires the computational hardness of McEP (see Section 3), which is typically implied by the computational hardness of McEPKP. However, Theorem 4 breaks this implication when McEPKP is instantiated using Raw Reed–Solomon codes.

To circumvent the above obstacle, we propose a new assumption, called the *Distorted McEliece problem* (DistMcEP). Along with this new assumption, we formalize its associated McEliece Public Key-style problem (DistMcEPKP), i.e., the distorted analogue of McEPKP.

Decisional Distorted McEliece assumption. When high structured codes are needed (as for achieving constant-rate robustness for a PRC), permuting the indexes (by right multiplying a generating matrix with a permutation matrix $\mathbf{P}$) is not enough to mask the structure of such structured codes. This potentially leads to distinguishing attacks using the dimension of the square code, as demonstrated in Theorem 4 with Raw Reed-Solomon codes.

The idea at the base of our new Distorted McEliece assumption is to replace the permutation matrix with a sparse invertible matrix from $\mathcal{SP}_{\gamma,n}(\mathbb{F}_q)$ (each

row has an integer constant $\gamma \geq 1$ of non-zero symbols). We call this matrix a distortion matrix of weight γ. This is a natural generalization of permutations, as the set of permutation matrices is exactly the set of sparse invertible matrices whose each row has Hamming weight 1.

We conjecture that a distortion matrix of weight $\gamma \geq 3$ is enough to circumvent distinguishing attacks on the square of Raw Reed-Solomon codes, while only decreasing the decoding capacity of the underlying error-correcting code of a factor γ, and thus impacting the associated robustness of the same factor, resulting in a constant-rate robust PRC. The choice of $\gamma \geq 3$ is driven by the fact that the polynomial attack of [COTG15] does not work when $\gamma > 2$ (see below).

Below, we formalize the Distorted McEliece assumption DistMcEP, which asks to distinguish between noisy codewords and uniformly random vectors, as in McEP, except that the public keys are distorted as we described above.

Definition 13 (Decisional Distorted McEliece Problem). *Let n, k be positive integers, q a prime power, $\eta \in (0,1)$, an integer $\gamma \geq 3$, and $\mathcal{F} \subset \mathbb{F}_q^{k \times n}$ a family of codes. Also, let $\mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F},\gamma}$ be a distribution that returns (distorted) samples generated as follows:*

1. *Sample a generating matrix $\mathbf{G} \hookleftarrow \mathcal{F}$.*
2. *Sample a change of basis matrix $\mathbf{S} \hookleftarrow GL_k(\mathbb{F}_q)$.*
3. *Sample a sparse invertible matrix $\mathbf{P} \hookleftarrow \mathcal{SP}_{\gamma,n}(\mathbb{F}_q) \cap \mathcal{GL}_n(\mathbb{F}_q)$.*[9]
4. *Output $\mathbf{G}' = \mathbf{SGP}^{-1}$.*

The Decisional Distorted McEliece Problem $\mathsf{DistMcEP}_{q,n,k,\mathcal{F},\gamma,\eta}$ asks to distinguish between distributions $(\mathbf{G}', \mathbf{xG}' + \mathbf{e})$ and $(\mathbf{G}', \mathbf{y})$, where $\mathbf{G}' \hookleftarrow \mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F},\gamma}$, $\mathbf{x} \hookleftarrow \mathbb{F}_q^k, \mathbf{e} \hookleftarrow \mathcal{B}_\eta(\mathbb{F}_q^n), \mathbf{y} \hookleftarrow \mathbb{F}_q^n$.

Decisional Distorted McEliece Public Key assumption. As for the case of McEP, we also introduce the distorted public-key version of McEPKP, named DistMcEPKP, which states that distorted public keys are indistinguishable from randomly sampled matrices. We provide the formal definition below.

Definition 14 (Decisional Distorted McEliece Public Key Problem). *Let n, k be positive integers, q a prime power, $\gamma \geq 3$, and $\mathcal{F} \subset \mathbb{F}_q^{k \times n}$ a family of codes. The Decisional Distorted Public Key Problem $\mathsf{DistMcEPKP}_{q,n,k,\mathcal{F},\gamma}$ asks to distinguish between distribution $\mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F},\gamma}$ and the uniform distribution over $\mathbb{F}_q^{k \times n}$.*

By using a hybrid argument, it is easy to see that computational hardness of DistMcEP (Definition 13) follows from the computational hardness of both DistMcEPKP (Definition 14) and LPN (the same way the computational hardness of standard McEP follows from that of McEPKP and LPN).

[9] This is the main difference between the distorted and the standard McEliece problem.

On the relationship between distorted and standard McEliece problems. A natural question is whether the distorted and standard McEliece problems are related. When considering the public-key McEliece problem, the relation is clear: the distorted problem DistMcEPKP is at least as hard as the standard McEPKP (when instantiated with the same family of underlying linear codes). This is because the public-key distribution of McEPKP uses permutation matrices, whereas the public-key distribution of DistMcEPKP uses sparse invertible matrices, which generalize permutation matrices. The following lemma formalizes the above hardness relation.

Theorem 7. *For any $\gamma \geq 1$, there exists a polynomial-time reduction from an instance of $\mathsf{McEPKP}_{q,n,k,\mathcal{F}}$ to an instance of $\mathsf{DistMcEPKP}_{q,n,k,\mathcal{F},\gamma}$.*

Proof. Let $\mathbf{G}'$ be an instance of $\mathsf{McEPKP}_{q,n,k,\mathcal{F}}$ (either $\mathbf{G}' = \mathbf{SGP}^{-1}$ with $\mathbf{P} \in \mathcal{S}_n$; or uniformly random). The reduction algorithm can simply sample a distortion matrix $\mathbf{Q} \in \mathcal{SP}_{\gamma,n}(\mathbb{F}_q) \cap \mathcal{GL}_n(\mathbb{F}_q)$ and define $\mathbf{G}'' = \mathbf{G}'\mathbf{Q}^{-1}$. The latter is an instance of $\mathsf{DistMcEPKP}_{q,n,k,\mathcal{F},\gamma}$:

- If $\mathbf{G}'$ was $\mathbf{SGP}^{-1}$, then $\mathbf{G}'' = \mathbf{SG}(\mathbf{QP})^{-1}$ and $\mathbf{QP}$ has naturally the same distribution as $\mathbf{Q}' \hookleftarrow \mathcal{SP}_{\gamma,n}(\mathbb{F}_q) \cap \mathcal{GL}_n(\mathbb{F}_q)$, hence $\mathbf{G}'' \hookleftarrow \mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F},\gamma}$;
- Otherwise, if $\mathbf{G}'$ was uniform, then $\mathbf{G}''$ is also uniform.

This concludes the proof. □

On the other hand, adapting the above reduction from McEP to DistMcEP (the non-public-key versions) is not possible. Intuitively, both DistMcEP and McEP require distinguishing noisy codewords from random ones, and the presence of such noise hinders any possibility of "transforming" the encoded public key of one assumption into the other without affecting the noise distribution. As a result, McEP and DistMcEP are not formally comparable. However, because the hardness of McEP and DistMcEP are usually proven thanks to the hardness of their respective public-key counterparts, given the above reduction (from McEPKP to DistMcEPKP), it is fair to say that DistMcEP is "somewhat" weaker than McEP when both are instantiated using the same code $\mathcal{C}$.

5.1 Evidence of Hardness of the Distorted McEliece Assumption with Raw Reed-Solomon codes

We now provide evidence supporting the plausible hardness of the newly proposed DistMcEP with Raw Reed-Solomon codes. This evidence is based on the following observations:

1. Couvreur *et al.* [COTG15] presented an attack against a McEliece encryption scheme where the public key is distorted using an approach similar to ours. We show that this attack does not extend to our distortion technique due to the higher distortion parameter considered. We also discuss the inapplicability of other attack techniques such as [CL21] and searching for small dual codewords in the public code (as for the family of polar codes [BCD+16]).

2. We conducted three experiments measuring the dimension of the square code associated with distorted Raw Reed-Solomon codes. Our experiments highlight the positive effect of our distortion technique in preventing distinguishers based on square code considerations, as the one presented in Section 4.

Resistance against the existing attack by Couvreur *et al.* [COTG15]. Our distortion is very similar to the work of [BBC+16] on McEliece encryption scheme, where the public key was masked by a distortion matrix $(\mathbf{R}+\mathbf{T})^{-1}$, where $\mathbf{R}$ is of rank 1 and $\mathbf{T}$ is sparse of weight γ. However, for efficiency reasons they were obliged to take $\gamma < 2$ which led to a polynomial attack [COTG15]. Let us explain this distinguishing attack and show why it does not apply to our case of $\gamma \geq 3$.

The attack proposed in [COTG15] makes extensive use of shortening of codes: for a given code $\mathcal{C} \subset \mathbb{F}_q^n$ and a subset $I \subset [1,n]$ of *positions*, the shortened code $Sh_I(\mathcal{C})$ is defined as follows:

$$Sh_I(\mathcal{C}) = \{(c_i)_{i\in[1,n]-I} \,|\, \exists \mathbf{c} = (c_i)_{i\in[1,n]} \text{ such that } \forall i \in I, c_i = 0\}.$$

The attack starts by shortening the dual of the public code $\mathcal{C}_{pub}^{\perp}$ ($\mathcal{C}_{pub}$ is the code generated by the distorted matrix $\mathbf{G}'$) on a set of random positions, until the square of the shortened code $Sh_I(\mathcal{C}_{pub}^{\perp})$ has a dimension lower than expected

$$\dim\left((Sh_I(\mathcal{C}_{pub}^{\perp}))^{\star 2}\right) < \min\left\{n-|I|, \binom{n-k-|I|+1}{2}\right\}.$$

Once such set of positions is found, the code $(Sh_I(\mathcal{C}_{pub}^{\perp})$ is then punctured in various positions in order to gain information on the sparse matrix $\mathbf{T}$.

For the attack to work, it is crucial that the set I of shortening positions is included in the set of indexes of rows of $\mathbf{T}$ of Hamming weight 1 (the so-called *positions of degree 1*), see [COTG15, Proposition 4]. In our construction, each row of $\mathbf{T}$ is of Hamming weight at least 3. Hence, the above attack cannot apply. The authors of [COTG15] clearly state that their techniques cannot be extended to work when the average Hamming weight of $\mathbf{T}$ is greater or equal to 2. We recommend $\gamma \geq 3$, so our parameters are way out of range of their attack.

Resistance against other attacks. Other possible attacks against our distorted McEliece assumption include the search for small dual codewords in the public code, as for the family of polar codes [BCD+16]. This is only possible when the dual contains polylogarithmic weights. As shown in [SKS19], Raw Reed-Solomon codes of dimension $k = n^{\alpha}$ for any constant $\alpha \in (0,1)$ have large dual distance $\Omega(\frac{n^{\alpha}}{\log n})$. The distortion will even amplify this effect and increase the dual distance by a factor γ, so attacks relying on the brute force search of dual codewords cannot target DistMcEP with Raw Reed-Solomon codes.

An encryption scheme based on permuted expanded Reed-Solomon codes (they are not Raw Reed-Solomon codes as they don't expand from a subcode but

rather from a full Reed-Solomon code) was proposed in [KRW21] and attacked in [CL21], so it is natural to consider whether the latter attack applies to our case. The attack from [CL21] crucially exploits the fact that the permutation from [KRW21] is actually a permutation of blocks, which preserves blocks of coordinates corresponding to the same initial coordinate in the extension field before expansion. As we consider generic permutations in our case, the attack from [CL21] cannot apply, even without distortion.

Experimental results on the square code dimension. In addition to the qualitative argument of resistance to previous attacks presented in the previous two paragraphs, we conducted three experiments to support our claims. For each experiment, we plot a comparison of the results between (standard) Raw Reed-Solomon codes and distorted Raw Reed-Solomon codes to show the effect of the distortion in preventing distinguishers based on square code considerations, such as the one presented in Section 4. Our SageMath script for running the experiments is submitted as an auxiliary artifact.

(a) *(Primal analysis)* For distorted Raw Reed-Solomon codes of small rates, we check that the dimension of the square code behaves like random codes. For a code $\mathcal{C}[N, K]$ with $N = nm$ and $K = \sqrt{N}$, we plot the difference

$$\Delta(N) = \binom{K+1}{2} - \dim(\mathcal{C}^{\star 2})$$

as a function of N. The choice of $K = \sqrt{N}$ is meant so as to have the maximal square code dimension $\binom{K+1}{2}$ below N. Therefore, $\Delta(N)$ is always ≥ 0 and an average of 0 for many codes $\mathcal{C}$ means that the dimension of $\mathcal{C}^{\star 2}$ experimentally behaves similarly to the dimension of the square of a random code. When $\Delta(N) > 0$, we call it a *loss of dimension* of the square code. We tested for $\mathcal{C} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x})$ (non-distorted case[10]) and for $\mathcal{C} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x}) \cdot \mathbf{P}^{-1}$ (distorted case), averaging the result over 100 different distortion matrices $\mathbf{P}$ sampled at random.

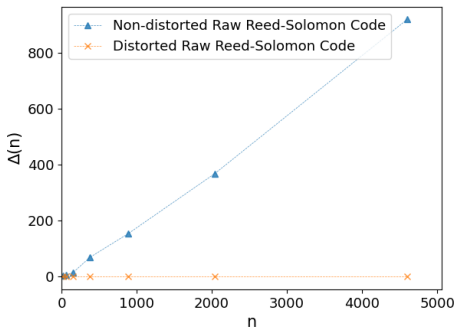
(b) *(Dual analysis)* For distorted Raw Reed-Solomon codes of large rates, we check that the dimension of the square code of the *dual* behaves like random codes. For a code $\mathcal{C}[N, K]$ such that $N - K = \sqrt{N}$, we plot the difference

$$\Delta(N) = \binom{N-K+1}{2} - \dim((\mathcal{C}^{\perp})^{\star 2})$$

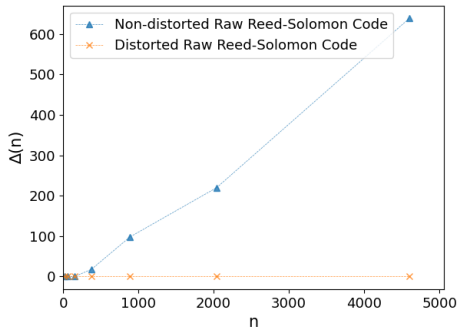
as a function of N. We have tested for $\mathcal{C} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x})$ (non-distorted case) and for $\mathcal{C} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x}) \cdot \mathbf{P}^{-1}$ (distorted case), averaging the result over 100 different distortion matrices $\mathbf{P}$ sampled at random.

(c) *(Dual shortening analysis)* For distorted Raw Reed-Solomon codes of half rates, we check that the dimension of the square code of shortenings of the

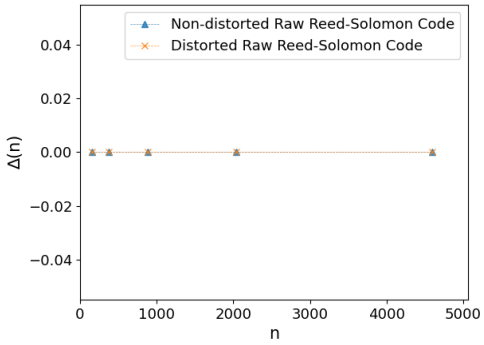
[10] It is useless to permute the code in the non-distorted case because square code dimensions are invariant by permutation



(a) Loss of dimension of the square of a non distorted (respectively distorted) Raw Reed-Solomon code of length n.



(b) Loss of dimension of the square of dual of a non distorted (respectively distorted) Raw Reed-Solomon code of length n.



(c) Loss of dimension of the square of shortening of the dual of non distorted (respectively distorted) Raw Reed-Solomon code of length n.

Fig. 2: Experimental results on the square code dimension of Raw Reed-Solomon codes.

dual behaves like random codes. For a code $\mathcal{C}[N, K]$ with $K = N/2$, choose a random subset of positions $I \subset [1, N]$, such that $\binom{N-K-|I|+1}{2} < N - |I|$, plot the difference

$$\Delta(N) = \binom{N - K - |I| + 1}{2} - \dim(Sh_I(\mathcal{C}^{\perp})^{\star 2})$$

as a function of N. We have tested for $\mathcal{C} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x})$ (non-distorted case) and for $\mathcal{C} = \mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x}) \cdot \mathbf{P}^{-1}$ (distorted case), averaging the result over 100 different distortion matrices $\mathbf{P}$ sampled at random.

Analysis of the results. The results of our experiments are presented in Figure 2. From experiments (a) and (b) (Figures 2a and 2b) we observe that $\Delta(N) \gg 1$ for non-distorted Raw Reed-Solomon codes, which is non surprising since it derives from the results of Section 4, and more precisely from Theorem 5. On the other hand, when applying a distortion to the Raw Reed-Solomon codes, $\Delta(N)$ averages to 0 both for the primal and the dual case, which gives experimental evidence that their square dimension behaves like random codes.

Experiment (c) aims at verifying that distorted Raw Reed-Solomon are not vulnerable to the attack from [COTG15], which uses squares of shortenings of the dual. It averages to 0, indicating that distorted Raw Reed-Solomon codes seem not vulnerable to this specific attack. As a side note, non-distorted Raw Reed-Solomon do not seem to be vulnerable either, and this is unsurprising since the [COTG15] attack does not specifically target this class of codes.

We selected our maximum values for N to reach relevant parameters for code-based pseudorandom encryption schemes with 128-bit security such as Classic McEliece ($N \approx 4000$) [ABC+22].

Remark 4 (On distorted LDPC codes). We have not considered DistMcEP with the family of LDPC, because for that family the distortion provably does not add any additional security, contrary to our conjecture for Raw Reed-Solomon codes. Indeed, the distortion of factor γ of an LDPC matrix of dual weight $O(\log n)$ will only multiply the dual weight by a factor γ, which as a result will stay in the class $O(\log n)$ and suffer from a quasipolynomial attack. Only codes with a dual weight of $O(n^\alpha)$ (such as Raw Reed-Solomon codes) can hope to have their security increased by a distortion.

6 A New Blueprint for Public-key PRCs

We adapt the PRC construction from Algorithm 1 to include a sparse invertible distortion matrix during key generation. This results in a PRC construction presented in Algorithm 2, whose pseudorandomness relies on DistMcEP (Definition 13) presented in the previous section. The only difference with Algorithm 1 is in the key generation where the permutation matrix $\mathbf{P}$ is replaced with a sparse invertible distortion matrix.

We could have also kept the initial blueprint exactly unchanged and have included the distortion matrix directly inside the family of codes, by defining a wider family $\mathcal{F}'$ of codes which contains codes from $\mathcal{F}$ multiplied by any distortion matrix $\mathbf{P}$. However, we prefer to put forward this new formulation, in order to highlight the role of the distortion matrix more precisely.

We refer the reader to Algorithm 2 for the formal specification, and, as for security, we provide the following result whose proof is deferred to Appendix B.3.

Theorem 8. *Let n, k be positive integers, q a prime power, $\eta \in (0,1)$, $\gamma \geq 3$, and $\mathcal{F} \subset \mathbb{F}_q^{k \times n}$ a family of codes that can bounded-distance correct $\delta \in (0, \frac{1}{2})$*

Algorithm 2 Substitution-robust public-key PRC from distorted codes

Parameters: A prime power q, a family $\mathcal{F}$ of linear $[n, k]$ codes in $\mathbb{F}_q$, a scrambling integer $\gamma \geq 3$, a noise parameter $\eta \in (0, 1)$, a correction parameter $\delta \in (0, \frac{1}{2})$.

Key-generation algorithm $\mathsf{KeyGen}(1^\lambda)$:

1. Sample a generating matrix $\mathbf{G} \hookleftarrow \mathcal{F}$ which defines a linear code $\mathcal{C}$ with decoding algorithm $\mathsf{Decode}_{\mathcal{C}}$.
2. Sample a change of basis matrix $\mathbf{S} \hookleftarrow GL_k(\mathbb{F}_q)$.
3. Sample a sparse invertible matrix $\mathbf{P} \hookleftarrow \mathcal{SP}_{\gamma,n}(\mathbb{F}_q) \cap \mathcal{GL}_n(\mathbb{F}_q)$.
4. Let $\mathbf{G}' = \mathbf{SGP}^{-1}$.
5. Sample $\mathbf{o} \hookleftarrow \mathbb{F}_q^n$.
6. Output $\mathsf{pk} = (\mathbf{G}', \mathbf{o})$ and $\mathsf{sk} = (\mathbf{P}, \mathbf{G}', \mathbf{o})$.

Encoding algorithm $\mathsf{Encode}(\mathsf{pk})$:

1. Let $\mathsf{pk} = (\mathbf{G}', \mathbf{o})$.
2. Sample $\mathbf{x} \hookleftarrow \mathbb{F}_q^k, \mathbf{e} \hookleftarrow \mathcal{B}_\eta(\mathbb{F}_q^n)$.
3. Output the codeword $\mathbf{y} = \mathbf{xG}' + \mathbf{e} + \mathbf{o}$.

Decoding algorithm $\mathsf{Decode}(\mathsf{sk}, \mathbf{y})$:

1. Let $\mathsf{sk} = (\mathbf{P}, \mathbf{G}', \mathbf{o})$.
2. Compute $\hat{\mathbf{y}} = \mathsf{Decode}_{\mathcal{C}}(\mathbf{y}')$ where $\mathbf{y}' = (\mathbf{y} - \mathbf{o})\mathbf{P}$.
3. If $\hat{\mathbf{y}} \in \mathcal{C}$ and $|\mathbf{y}' - \hat{\mathbf{y}}| \leq \delta n$, **return** $\top$.
4. Otherwise, **return** $\bot$.

errors efficiently, such that $n(1 - H_q(\delta)) - k = \Omega(n)$. *Assuming hardness of the Distorted McEliece Problem* $\mathsf{DistMcEP}_{q,n,k,\mathcal{F},\gamma,\eta}$ *for the family* $\mathcal{F}$*, the PRC from Algorithm 2 is pseudorandom, sound and has strong adaptive robustness to any channel introducing at most* $\delta' = \frac{\delta}{\gamma} - 2\eta$ *fraction of substitutions.*

Remark 5. (On choice of parameters) To get a public-key PRC with soundness, pseudorandomness, and robustness up to $\delta' = \frac{1}{2\gamma} - \varepsilon$ for any small constant $\varepsilon > 0$, we can use the family of Raw Reed-Solomon codes of dimension $k = \Theta(n^\alpha) = o(n)$ for some constant $\alpha \in (0, 1)$, and choose $\delta = \frac{1}{2} - \varepsilon'$ for some small constant $\varepsilon' > 0$ which depends only on ε and γ. This means we can reach a maximal robustness of $\frac{1}{6}$ for our minimal suggestion of parameter $\gamma = 3$.

7 Multi-bit PRCs with Strong Adaptive Robustness

We start by extending the definitions of zero-bit PRCs to the multi-bit case, i.e., PRCs that encode arbitrary messages.

Definition 15 (Multi-bit public-key PRCs). *Let* Σ *be an alphabet. A (multi-bit) public-key PRC of message length* k *and codeword length* n *with strong adaptive* δ*-robustness is described by a triple of efficient randomized algorithms*

$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$, $\mathsf{Encode}(\mathsf{pk}, \mathbf{m} \in \Sigma^k) \to \Sigma^n$, $\mathsf{Decode}(\mathsf{sk}, \mathbf{y} \in \Sigma^n) \to \Sigma^k \cup \{\perp\}$, *satisfying the following properties:*

- *(Computational Pseudorandomness) For any probabilistic polynomial-time adversary $\mathcal{A}$, we have:*

$$\left|\mathbb{P}[\mathcal{A}^{\mathsf{Encode}(\mathsf{pk},\cdot)}(1^\lambda, \mathsf{pk}) = 1] - \mathbb{P}[\mathcal{A}^{\mathcal{U}}(1^\lambda, \mathsf{pk}) = 1]\right| \leq \mathsf{negl}(\lambda),$$

  where the probabilities are taken over the random coins of $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $\mathcal{A}$. The notation $\mathcal{A}^{\mathcal{U}}$ means that $\mathcal{A}$ has access to an oracle that responds to each query with a freshly uniform random value in Σ^n.
- *(Soundness) For any fixed $\mathbf{c} \in \Sigma^n$, we have:*

$$\mathbb{P}_{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)}[\mathsf{Decode}(\mathsf{sk}, \mathbf{c}) = \perp] \geq 1 - \mathsf{negl}(\lambda).$$

- *(Strong Adaptive δ-robustness) For any probabilistic polynomial-time adversary $\mathcal{A}$, we have:*

$$\mathbb{P}_{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)}\left[\begin{array}{l|l} \mathsf{Decode}(\mathsf{sk}, \mathbf{y}) \neq \mathbf{m} & \mathbf{m} \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk}) \\ d(\mathbf{c}, \mathbf{y}) \leq \delta n & \mathbf{c} \leftarrow \mathsf{Encode}(\mathsf{pk}, \mathbf{m}) \\ & \mathbf{y} \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk}, \mathbf{c}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Remark 6. The experiment for strong adaptive robustness in the multi-bit setting may seem weaker than the definition of (non-strong) adaptive robustness from [AAC+25] where the attacker gets to choose the internal randomness of the codeword $\mathbf{c}$. We made this choice of definition to remain consistent with the definition of strong adaptive robustness in the zero-bit setting from Section 2.3 (which is also the definition from [CGG+26]). We point out that, however, all our constructions in this paper (both in the zero-bit and multi-bit setting) would also achieve strong adaptive robustness when the adversary gets to choose the randomness. This is because the Reed-Solomon decoder is able to correct any worst-case error for any codeword, regardless of how it was generated (with honest or adversarial randomness).

7.1 Multi-bit PRC for uniformly random messages

In this section, we provide a construction of a multi-bit PRC whose pseudorandomness property holds only when randomly sampled messages are encoded. We call this pseudorandomness property *weak pseudorandomness*.

Definition 16 (Weak pseudorandomness). *A multi-bit public-key PRC* $(\mathsf{KeyGen}, \mathsf{Encode}, \mathsf{Decode})$ *is weakly pseudorandom if for any probabilistic polynomial-time adversary $\mathcal{A}$, we have:*

$$\left|\mathbb{P}[\mathcal{A}^{\mathcal{O}}(1^\lambda, \mathsf{pk}) = 1] - \mathbb{P}[\mathcal{A}^{\mathcal{U}}(1^\lambda, \mathsf{pk}) = 1]\right| \leq \mathsf{negl}(\lambda),$$

where the probabilities are taken over the random coins of $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $\mathcal{A}$. The notation $\mathcal{A}^{\mathcal{O}}$ means that $\mathcal{A}$ has access to an oracle that responds to each query by sampling a uniform random message $\mathbf{m} \in \Sigma^k$ and then responding with $\mathsf{Encode}(\mathsf{pk}, \mathbf{m})$; $\mathcal{U}$ denotes an oracle that responds with a uniform value in Σ^n.

We also need to change the robustness game: the adversary does not get to choose the message $\mathbf{m}$ arbitrarily, but is provided with a random $\mathbf{x}$ before answering the randomness and the modified codeword.

Definition 17 (Strong adaptive robutness for uniform messages). *A multi-bit public-key PRC* $(\mathsf{KeyGen}, \mathsf{Encode}, \mathsf{Decode})$ *for uniformly random messages is strongly adaptive δ-robust if for any probabilistic polynomial-time adversary $\mathcal{A}$, we have:*

$$\mathbb{P}_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)}\left[\begin{array}{l|l}\mathsf{Decode}(\mathsf{sk},\mathbf{y})\neq\mathbf{m} & \mathbf{m}\hookleftarrow\Sigma^k\\ d(\mathbf{c},\mathbf{y})\leq\delta n & \mathbf{c}\leftarrow\mathsf{Encode}(\mathsf{pk},\mathbf{m})\\ & \mathbf{y}\leftarrow\mathcal{A}(\mathbf{c},\mathbf{m},\mathsf{pk},\mathsf{sk})\end{array}\right]\leq\mathsf{negl}(\lambda).$$

In order to be able to decode multi-bit messages, we must use a stronger algorithm than a bounded-distance decoder; we need a unique decoder. We will denote a unique decoder $\mathsf{UDecode}$ to differentiate from previous bounded-distance decoders.

Definition 18 (Unique decoding). *A code $\mathcal{C}$ satisfies the unique decoding property up to a fraction $\delta\in(0,1)$ of errors if there exists a PPT algorithm $\mathsf{UDecode}_\mathcal{C}:\mathbb{F}_q^n\to\mathbb{F}_q^n$ such that for any $\mathbf{c}\in\mathcal{C}$ and $\mathbf{y}\in\mathbb{F}_q^n$ such that $d(\mathbf{y},\mathbf{c})\leq\delta n$, we have $\mathbf{c}=\mathsf{UDecode}_\mathcal{C}(\mathbf{y})$ with probability $\geq 1-2^{-\Omega(n)}$ over the randomness of $\mathsf{UDecode}_\mathcal{C}$.*

Thanks to their large minimal distance and list decoding algorithm, the Raw Reed-Solomon codes support unique decoding up to almost $1/4$ fraction of errors.

Proposition 7 ([SKS19, Thm. 3.1]). *For any constants $\varepsilon>0,\alpha\in(0,1)$, for a sufficiently large n, with $k=n^\alpha$, and for any support $\mathbf{x}$, the code $\mathcal{C}=\mathsf{SimpleRawRS}_{k,\mathcal{B}}(\mathbf{x})$ is uniquely decodable from a fraction of $\frac{1}{4}-\varepsilon$ errors in the sense of Definition 18.*

We can now present, in Algorithm 3, our construction for a multi-bit public-key PRC for uniformly random messages. In comparison with Algorithm 2, in the encoding algorithm, the vector $\mathbf{x}$ is not part of the internal randomness anymore, but is a random message to be encoded. To recover exactly $\mathbf{x}$ in the decoding algorithm, we therefore need a unique decoder for $\mathcal{C}$, which halves the maximal decoding capacity (and the robustness). The following theorem states the property of this new PRC.

Theorem 9. *Let n,k be positive integers, q a prime power, $\eta\in(0,1)$, $\gamma\geq 3$, and $\mathcal{F}\subset\mathbb{F}_q^{k\times n}$ a family of codes that can bounded-distance correct $\delta\in(0,\frac{1}{4})$ errors efficiently, such that $n(1-H_q(\delta))-k=\Omega(n)$. Assuming hardness of the Distorted McEliece Problem $\mathsf{DistMcEP}_{q,n,k,\mathcal{F},\gamma,\eta}$ for the family $\mathcal{F}$, the PRC from Algorithm 3 is pseudorandom, sound and has strong adaptive robustness to any channel introducing at most $\delta'=\frac{\delta}{\gamma}-2\eta$ fraction of substitutions.*

Algorithm 3 Substitution-robust public-key PRC from distorted codes for multi-bit uniformly random messages

Parameters: A prime power q, a family $\mathcal{F}$ of linear $[n, k]$ codes in $\mathbb{F}_q$, a scrambling integer $\gamma \geq 3$, a noise parameter $\eta \in (0, 1)$, a correction parameter $\delta \in (0, \frac{1}{4})$.

Key-generation algorithm $\mathsf{KeyGen}(1^\lambda)$:

1. Sample a generating matrix $\mathbf{G} \hookleftarrow \mathcal{F}$ which defines a linear code $\mathcal{C}$ with decoding algorithm $\mathsf{UDecode}_{\mathcal{C}}$.
2. Sample a change of basis matrix $\mathbf{S} \hookleftarrow GL_k(\mathbb{F}_q)$.
3. Sample a sparse invertible matrix $\mathbf{P} \hookleftarrow \mathcal{SP}_{\gamma,n}(\mathbb{F}_q) \cap \mathcal{GL}_n(\mathbb{F}_q)$.
4. Let $\mathbf{G}' = \mathbf{SGP}^{-1}$.
5. Sample $\mathbf{o} \hookleftarrow \mathbb{F}_q^n$.
6. Output $\mathsf{pk} = (\mathbf{G}', \mathbf{o})$ and $\mathsf{sk} = (\mathbf{P}, \mathbf{G}', \mathbf{o})$.

Encoding algorithm $\mathsf{Encode}(\mathsf{pk}, \mathbf{x} \hookleftarrow \mathbb{F}_q^k)$:

1. Let $\mathsf{pk} = (\mathbf{G}', \mathbf{o})$.
2. Sample $\mathbf{e} \hookleftarrow \mathcal{B}_\eta(\mathbb{F}_q^n)$.
3. Output the codeword $\mathbf{y} = \mathbf{x}\mathbf{G}' + \mathbf{e} + \mathbf{o}$.

Decoding algorithm $\mathsf{Decode}(\mathsf{sk}, \mathbf{y})$:

1. Let $\mathsf{sk} = (\mathbf{P}, \mathbf{G}', \mathbf{o})$.
2. Compute $\hat{\mathbf{y}} = \mathsf{UDecode}_{\mathcal{C}}(\mathbf{y}')$ where $\mathbf{y}' = (\mathbf{y} - \mathbf{o})\mathbf{P}$.
3. If $\hat{\mathbf{y}} \in \mathcal{C}$ and $|\mathbf{y}' - \hat{\mathbf{y}}| \leq \delta n$, **return** the unique $\hat{\mathbf{x}} \in \mathbb{F}_q^k$ such that $\hat{\mathbf{x}}\mathbf{G}' = \hat{\mathbf{y}}\mathbf{P}^{-1}$.
4. Otherwise, **return** $\perp$.

Proof. Because the output distributions of Encode are exactly the same between Algorithm 2 and Algorithm 3, the weak pseudorandomness of the multi-bit PRC property follows immediately from the pseudorandomness of the zero-bit PRC. The same goes for soundness since the conditions for outputting $\perp$ are identical between the two constructions. Therefore, we only need to prove the robustness with the following lemma. □

Lemma 5 (Strong adaptive robustness). *If any code from $\mathcal{F}$ can uniquely correct δ errors efficiently, then the PRC from Algorithm 3 is strongly adaptively robust against any channel introducing a fraction of at most $\delta' = \frac{\delta}{\gamma} - 2\eta$ substitutions.*

Proof. Let n be polynomial in λ. Fix any keypair $(\mathsf{pk} = (\mathbf{G}', \mathbf{o}), \mathsf{sk})$, where $\mathbf{G}' = \mathbf{SGP}^{-1}$. Sample a random message $\mathbf{x} \hookleftarrow \mathbb{F}_q^k$, and let $\mathbf{c} = \mathsf{Encode}(\mathsf{pk}, \mathbf{x})$. Let $\mathbf{y}$ be the modified codeword output by the adversary. We need to show that if $d(\mathbf{c}, \mathbf{y}) \leq \delta' n$, then $\mathsf{Decode}(\mathsf{sk}, \mathbf{y}) = \mathbf{x}$ except with negligible probability.

Let $\mathbf{y}' = (\mathbf{y} - \mathbf{o})\mathbf{P}$ computed in the first step of Decode. Using the proof of Lemma 11, we have that $d(\mathbf{y}', \mathbf{xSG}) \leq \delta n$.

Algorithm 4 Public-key PRC for multi-bit arbitrary messages

Parameters: A multi-bit PRC for uniformly random messages in $\mathbb{F}_q^k$ $(\mathsf{PRC}_{\mathsf{rand}}.\mathsf{KeyGen}, \mathsf{PRC}_{\mathsf{rand}}.\mathsf{Encode}, \mathsf{PRC}_{\mathsf{rand}}.\mathsf{Decode})$ and a public-key encryption scheme with pseudorandom ciphertexts $(\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ with plaintext space $\mathbb{F}_q^\ell$ and ciphertext space $\mathbb{F}_q^k$.

Key-generation algorithm $\mathsf{KeyGen}(1^\lambda)$:

1. $(\mathsf{pk}_{\mathsf{PRC}}, \mathsf{sk}_{\mathsf{PRC}}) = \mathsf{PRC}_{\mathsf{rand}}.\mathsf{KeyGen}(1^\lambda)$.
2. $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) = \mathsf{PKE.KeyGen}(1^\lambda)$.
3. **return** $(\mathsf{pk}, \mathsf{sk}) = ((\mathsf{pk}_{\mathsf{PRC}}, \mathsf{pk}_{\mathsf{PKE}}), (\mathsf{sk}_{\mathsf{PRC}}, \mathsf{sk}_{\mathsf{PKE}}))$.

Encoding algorithm $\mathsf{Encode}(\mathsf{pk} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}), \mathbf{m} \in \mathbb{F}_q^\ell)$:

1. $\mathbf{x} = \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{m})$
2. **return** $\mathsf{PRC}_{\mathsf{rand}}.\mathsf{Encode}(\mathsf{pk}_{\mathsf{PRC}}, \mathbf{x})$

Decoding algorithm $\mathsf{Decode}(\mathsf{sk} = (\mathsf{sk}_{\mathsf{PRC}}, \mathsf{sk}_{\mathsf{PKE}}), \mathbf{y})$:

1. $\mathbf{x} = \mathsf{PRC}_{\mathsf{rand}}.\mathsf{Decode}(\mathsf{sk}_{\mathsf{PRC}}, \mathbf{y})$.
2. If $\mathbf{x} = \perp$, return $\perp$
3. Otherwise, **return** $\mathsf{PKE.Dec}(\mathsf{sk}_{\mathsf{PKE}}, \mathbf{x})$.

By the unicity property of the decoder, we are now sure, except with negligible probability, that the output $\hat{\mathbf{y}}$ of $\mathsf{UDecode}_{\mathcal{C}}(\mathbf{y}')$ is exactly $\hat{\mathbf{y}} = \mathbf{xSG}$. As a result, we have $\hat{\mathbf{y}}\mathbf{P}^{-1} = \mathbf{xSGP}^{-1} = \mathbf{xG}'$.

Because $\mathbf{G}'$ is a full-rank matrix, $\mathbf{x}$ is therefore the unique $\hat{\mathbf{x}}$ such that $\hat{\mathbf{x}}\mathbf{G} = \hat{\mathbf{y}}\mathbf{P}^{-1}$. This shows that the PRC Decode algorithm will return exactly $\mathbf{x}$ except with negligible probability. □

7.2 Multi-bit PRC for arbitrary messages

In this section, we present a transformation to upgrade a multi-bit PRC for random messages into a multi-bit PRC for arbitrary messages, using a public-key encryption scheme with pseudorandom ciphertexts. Since we don't need any robustness for the PKE, we can use the McEliece cryptosystem with Goppa codes, which already has pseudorandom ciphertexts under the corresponding McEliece assumption. As a result, we don't need an additional security assumption. The syntax and security notions of PKE with pseudorandom ciphertexts are defined in Appendix A.

Theorem 10. *Let* $\mathsf{PRC}_{\mathsf{rand}}$ *be a multi-bit PRC construction which is sound, weakly pseudorandom, strongly adaptive against any* δ *fraction of substitutions, supporting uniformly random messages, and* PKE *be a public-key encryption scheme with pseudorandom ciphertexts. Then the* PRC *from Algorithm 4 is a multi-bit PRC supporting arbitrary messages, which is sound, pseudorandom under the same assumptions as* $\mathsf{PRC}_{\mathsf{rand}}$ *and* PKE*, and strongly adaptive against the same* δ *fraction of substitutions.*

Proof. Let us deal with the properties separately.

- *Soundness*: $\mathsf{PRC}.\mathsf{Decode}$ returns $\perp$ if and only if $\mathsf{PRC}_{\mathsf{rand}}.\mathsf{Decode}$ returns $\perp$ with the same input vector $\mathbf{y}$, so the soundness of $\mathsf{PRC}_{\mathsf{rand}}$ directly transfers to PRC.
- *Pseudorandomness*: We will consider a simple hybrid game (for the pseudorandomness of the PRC), which is defined by modifying the step 1. of the encoding by replacing $\mathbf{x}$ with uniform randomness: $\mathbf{x} \leftarrow\!\!\$\ \mathbb{F}_q^k$. The real pseudorandomness game is indistinguishible from the hybrid game because the ciphertext of PKE are pseudorandom. Now, the hybrid game is exactly the game for the weak pseudorandomness of $\mathsf{PRC}_{\mathsf{rand}}$ because the Encode algorithm of PRC now returns exactly $\mathsf{Encode}(\mathsf{pk}, \mathbf{x} \leftarrow\!\!\$\ \mathbb{F}_q^k)$.
- *Robustness*: A PRC codeword $\mathbf{y}$ is also a codeword $\mathsf{PRC}_{\mathsf{rand}}$ (although for a different message), so if $\mathbf{y}$ has been modified in less than δn positions, by the strong adaptive robustness property of $\mathsf{PRC}_{\mathsf{rand}}$, the recovered $\mathbf{x} = \mathsf{PRC}_{\mathsf{rand}}.\mathsf{Decode}(\mathsf{sk}, \mathbf{y})$ will be exactly the one from the encoding algorithm, of the form $\mathbf{x} = \mathsf{Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{m})$, and we will have $\mathsf{PKE}.\mathsf{Dec}(\mathsf{sk}_{\mathsf{PKE}}, \mathbf{x}) = \mathbf{m}$.

This finishes the proof. □

Acknowledgments

Danilo Francati was supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union-NextGenerationEU. Daniele Venturi is member of the Gruppo Nazionale Calcolo Scientifico Istituto Nazionale di Alta Matematica (GNCS-INdAM). His research was partially supported by project SERICS (PE00000014) and by project PARTHENON (B53D23013000006)—under the MUR National Recovery and Resilience Plan funded by the European Union-NextGenerationEU—by project BEAT—funded by Sapienza University of Rome—and by the AI-Sec Lab initiative between Sapienza University of Rome and the CYBER 4.0 Competence Center. This work was done while Victor Dyseryn was a visitor at Sapienza University of Rome.

References

AAC+25. Omar Alrabiah, Prabhanjan Ananth, Miranda Christ, Yevgeniy Dodis, and Sam Gunn. Ideal pseudorandom codes. In Michal Koucký and Nikhil Bansal, editors, *57th ACM STOC*, pages 1638–1647. ACM Press, June 2025.

ABC+22. Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic McEliece. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions.

BBC+16. Marco Baldi, Marco Bianchi, Franco Chiaraluce, Joachim Rosenthal, and Davide Schipani. Enhanced public key security for the McEliece cryptosystem. *Journal of Cryptology*, 29(1):1–27, January 2016.

BCD+16. Magali Bardet, Julia Chaulet, Vlad Dragoi, Ayoub Otmani, and Jean-Pierre Tillich. Cryptanalysis of the McEliece public key cryptosystem based on polar codes. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, pages 118–143. Springer, Cham, 2016.

BL23. Pierre Briaud and Pierre Loidreau. Cryptanalysis of rank-metric schemes based on distorted gabidulin codes. In Thomas Johansson and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 14th International Workshop, PQCrypto 2023*, pages 38–56. Springer, Cham, August 2023.

BMT23. Magali Bardet, Rocco Mora, and Jean-Pierre Tillich. Polynomial time key-recovery attack on high rate random alternant codes. *IEEE Transactions on Information Theory*, 70(6):4492–4511, 2023.

CC20. Daniel Coggia and Alain Couvreur. On the security of a loidreau rank metric code based encryption scheme. *DCC*, 88(9):1941–1957, 2020.

CCMZ15. Ignacio Cascudo, Ronald Cramer, Diego Mirandola, and Gilles Zémor. Squares of random linear codes. *IEEE Transactions on Information Theory*, 61(3):1159–1173, 2015.

CG24. Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VI*, volume 14925 of *LNCS*, pages 325–347. Springer, Cham, August 2024.

CGG+14. Alain Couvreur, Philippe Gaborit, Valérie Gauthier-Umaña, Ayoub Otmani, and Jean-Pierre Tillich. Distinguisher-based attacks on public-key cryptosystems using reed-solomon codes. *DCC*, 73(2):641–666, 2014.

CGG+26. Miranda Christ, Noah Golowich, Sam Gunn, Ankur Moitra, and Daniel Wichs. Improved pseudorandom codes from permuted puzzles. *to appear in STOC 2026*, 2026.

CGZ24. Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In Shipra Agrawal and Aaron Roth, editors, *The Thirty Seventh Annual Conference on Learning Theory, June 30 - July 3, 2023, Edmonton, Canada*, Proceedings of Machine Learning Research, pages 1125–1139. PMLR, 2024.

CL21. Alain Couvreur and Matthieu Lequesne. On the security of subspace subcodes of reed–solomon codes for public key encryption. *IEEE Transactions on Information Theory*, 68(1):632–648, 2021.

CMCP14. Alain Couvreur, Irene Márquez-Corbella, and Ruud Pellikaan. A polynomial time attack against algebraic geometry code based public key cryptosystems. In *2014 IEEE International Symposium on Information Theory*, pages 1446–1450. IEEE, 2014.

COT16. Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. Polynomial time attack on wild mceliece over quadratic extensions. *IEEE Transactions on Information Theory*, 63(1):404–427, 2016.

COTG15. Alain Couvreur, Ayoub Otmani, Jean-Pierre Tillich, and Valérie Gauthier-Umaña. A polynomial-time attack on the BBCRS scheme. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 175–193. Springer, Berlin, Heidelberg, March / April 2015.

DMR25. Nico Döttling, Anne Müller, and Mahesh Sreekumar Rajasree. Separating pseudorandom codes from local oracles. In Benny Applebaum and

Huijia (Rachel) Lin, editors, *TCC 2025, Part IV*, volume 16271 of *LNCS*, pages 225–257. Springer, Cham, December 2025.

FGP+25. Danilo Francati, Yevin Nikhel Goonatilake, Shubham Pawar, Daniele Venturi, and Giuseppe Ateniese. The coding limits of robust watermarking for generative models. Cryptology ePrint Archive, Report 2025/1620, 2025.

FGUO+13. Jean-Charles Faugere, Valérie Gauthier-Umana, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high-rate mceliece cryptosystems. *IEEE Transactions on Information Theory*, 59(10):6830–6844, 2013.

Gal62. Robert Gallager. Low-density parity-check codes. *IRE Transactions on information theory*, 8(1):21–28, 1962.

GG25. Surendra Ghentiyala and Venkatesan Guruswami. New constructions of pseudorandom codes. In Alina Ene and Eshan Chattopadhyay, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2025, Berkeley, CA, USA, August 11-13, 2025*, pages 54:1–54:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

GGW25. Sanjam Garg, Sam Gunn, and Mingyuan Wang. Black-box crypto is useless for pseudorandom codes. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025, Part IV*, volume 16271 of *LNCS*, pages 205–224. Springer, Cham, December 2025.

GM24. Noah Golowich and Ankur Moitra. Edit distance robust watermarks via indexing pseudorandom codes. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

KGW+23. John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 17061–17084. PMLR, 2023.

KRW21. Karan Khathuria, Joachim Rosenthal, and Violetta Weger. Encryption scheme based on expanded reed-solomon codes. *Advances in Mathematics of Communications*, 15(2):207–218, 2021.

KTHL24. Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Trans. Mach. Learn. Res.*, 2024, 2024.

Loi17. Pierre Loidreau. A new rank metric codes based encryption scheme. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, pages 3–17. Springer, Cham, 2017.

Loi22. Pierre Loidreau. Analysis of a public-key encryption scheme based on distorted gabidulin codes. In *Proceedings of the twelth international workshop on coding and cryptography WCC*, 2022.

McE78a. RJ McEliece. A public-key system based on algebraic coding theory, 114-116. deep sace network progress report, 44. *Jet Propulsion Laboratory, California Institute of Technology*, 3:85–86, 1978.

McE78b. Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. The deep space network progress report 42-44, Jet Propulsion Laboratory, California Institute of Technology, January/February 1978. https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF.

MT23. Rocco Mora and Jean-Pierre Tillich. On the dimension and structure of the square of the dual of a goppa code. *DCC*, 91(4):1351–1372, 2023.

SKS19. Jad Silbak, Swastik Kopparty, and Ronen Shaltiel. Quasilinear time list-decodable codes for space bounded channels. In David Zuckerman, editor, *60th FOCS*, pages 302–333. IEEE Computer Society Press, November 2019.

SS92. Vladimir Michilovich Sidelnikov and Sergey O Shestakov. On insecurity of cryptosystems based on generalized reed-solomon codes. 1992.

ZALW24. Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

ZEF+24. Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 58851–58880. PMLR / OpenReview.net, 2024.

A PKE with Pseudorandom Ciphertexts

We recall the standard notion of a public-key encryption (PKE) scheme with pseudorandom ciphertexts.

Definition 19 (Public-key encryption with pseudorandom ciphertexts). *Let Σ be an alphabet. A PKE with pseudorandom ciphertexts is described by a triple of efficient randomized algorithms $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$, $\mathsf{Enc}(\mathsf{pk}, \mathbf{m} \in \Sigma^*) \to \Sigma^*$, $\mathsf{Dec}(\mathsf{sk}, \mathbf{c} \in \Sigma^*) \to \Sigma^*$, satisfying the following properties:*

- *(Pseudorandomness) For any probabilistic polynomial-time adversary $\mathcal{A}$, we have:*
$$\left|\mathbb{P}[\mathcal{A}^{\mathsf{Enc}(\mathsf{pk},\cdot)}(1^\lambda, \mathsf{pk}) = 1] - \mathbb{P}[\mathcal{A}^{\mathcal{U}}(1^\lambda, \mathsf{pk}) = 1]\right| \leq \mathsf{negl}(\lambda),$$
where the probabilities are taken over the random coins of $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $\mathcal{A}$. The notation $\mathcal{A}^{\mathcal{U}}$ means that $\mathcal{A}$ has access to an oracle that responds to each query with a freshly uniform random value in Σ^*.
- *(Correctness) For any message $\mathbf{m} \in \Sigma^*$,*
$$\mathbb{P}_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)}[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathbf{m})) = \mathbf{m}] \geq 1 - \mathsf{negl}(\lambda).$$

B Supporting Proofs

B.1 Proof of Proposition 4

By leveraging [SKS19, Lemma 3.3], we obtain that, when the maximum degree of evaluated polynomials is small enough (the latter is ensured by our condition $k = n^\alpha$ for large n), SimpleRawRS codes are list-decodable from a fraction of $\frac{1}{2} - \varepsilon$ with a list of size $O(n^2)$. We now show that the list-decodability property implies the notion of bounded-distance decodability (Definition 4).

List-decodability for a list of size L ensures that for any $\mathbf{y}$ such that $d(\mathbf{y}, \mathcal{C}) \leq \delta n$ (i.e. there exists $\hat{\mathbf{y}} \in \mathcal{C}$ with $|\mathbf{y} - \hat{\mathbf{y}}| \leq \delta n$), then the algorithm $\mathsf{Decode}_{\mathcal{C}}^{\mathrm{list}}(\mathbf{y})$ returns a list ℓ of size at most L such that $\hat{\mathbf{y}} \in \ell$.

A valid bounded-distance algorithm $\mathsf{Decode}_{\mathcal{C}}$ can be obtained by searching in the list output by $\mathsf{Decode}_{\mathcal{C}}^{\mathrm{list}}(\mathbf{y})$ an element $\hat{\mathbf{y}}$ such that $\hat{\mathbf{y}} \in \mathcal{C}$ and $|\mathbf{y} - \hat{\mathbf{y}}| \leq \delta n$, which is guaranteed to happen (thanks to the property of $\mathsf{Decode}_{\mathcal{C}}^{\mathrm{list}}$) in polynomial time, since the size of the list is polynomial.

B.2 Proof of Theorem 3

We prove each PRC property of Algorithm 1 individually in Lemmas 6 to 8, and the corresponding proofs are similar to those in [CG24].

Lemma 6 (Soundness). *For a family $\mathcal{F}$ of linear codes of dimension k, whenever $n(1 - H_q(\delta)) - k = \Omega(n)$, the PRC from Algorithm 1 is sound.*

Proof. Let $\mathbf{y} \in \mathbb{F}_q^n$ be a fixed vector. For soundness, because n is a polynomial in λ, it is sufficient to prove

$$\mathbb{P}_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)}\left[\mathsf{Decode}(\mathsf{sk},\mathbf{y}) = \top\right] \leq \mathsf{negl}(n).$$

Since when $\mathbf{P} \hookleftarrow \mathcal{S}_n$ and $\mathbf{o} \hookleftarrow \mathbb{F}_q^n$, $(\mathbf{y}-\mathbf{o})\mathbf{P} \hookleftarrow \mathbb{F}_q^n$, this is equivalent to

$$\mathbb{P}_{\mathbf{y}'\hookleftarrow\mathbb{F}_q^n}\left[\exists \hat{\mathbf{y}} \in \mathcal{C}, |\mathbf{y}' - \hat{\mathbf{y}}| \leq \delta n\right] \leq \mathsf{negl}(n).$$

We observe that $\exists \hat{\mathbf{y}} \in \mathcal{C}, |\mathbf{y}' - \hat{\mathbf{y}}| \leq \delta n \iff \mathbf{y}' \in \bigcup_{\hat{\mathbf{y}}\in\mathcal{C}} \mathcal{B}(\hat{\mathbf{y}}, \delta n)$. Now, a standard estimate for the volume of a Hamming ball gives

$$\sharp\left\{\bigcup_{\hat{\mathbf{y}}\in\mathcal{C}} \mathcal{B}(\hat{\mathbf{y}}, \delta n)\right\} \leq q^{k+H_q(\delta)n+o(n)}.$$

Hence, the following inequality holds:

$$\mathbb{P}_{\mathbf{y}'\hookleftarrow\mathbb{F}_q^n}\left[\exists \hat{\mathbf{y}} \in \mathcal{C}, |\mathbf{y}' - \hat{\mathbf{y}}| \leq \delta n\right] \leq q^{k-n+H_q(\delta)n+o(n)} = q^{k-n(1-H_q(\delta))+o(n)},$$

where the latter is a negligible quantity whenever $n(1-H_q(\delta)) - k = \Omega(n)$. □

Lemma 7 (Pseudorandomness). *The pseudorandomness of the PRC of Algorithm 1 follows from the hardness of* $\mathsf{McEP}_{q,n,k,\mathcal{F},\eta}$.

Proof. Consider any polynomial-time algorithm $\mathcal{A}$ which satisfies

$$\left|\mathbb{P}_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)}[\mathcal{A}^{\mathsf{Encode}(\mathsf{pk})}(\mathsf{pk}) = 1] - \mathbb{P}_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)}[\mathcal{A}^{\mathcal{U}}(\mathsf{pk}) = 1]\right| = \nu(\lambda)$$

We will use $\mathcal{A}$ to build a polynomial-time adversary $\mathcal{A}'$ which solves $\mathsf{McEP}_{q,n,k,\mathcal{F},\eta}$ with advantage ν.

On each sample $(\mathbf{G},\mathbf{y})$ with $\mathbf{G} \hookleftarrow \mathcal{D}^{\mathsf{public}}_{q,n,k,\mathcal{F}}$ and $\mathbf{y}$ being either uniform or $\mathbf{xG}+\mathbf{e}$, $\mathcal{A}'$ can simulate the oracle call of $\mathcal{A}$ (either $\mathsf{Encode}(\mathsf{pk})$ or $\mathcal{U}$) by sampling $\mathbf{o} \hookleftarrow \mathbb{F}_q^n$ and outputting $\mathbf{y}+\mathbf{o}$.

$\mathcal{A}'$ can now query algorithm $\mathcal{A}$ with input $\mathsf{pk} = (\mathbf{G},\mathbf{o})$ and simulating the oracle as described above.

It clearly runs in polynomial time, and because $(\mathbf{G},\mathbf{o})$ defined above has exactly the same distribution as pk in $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ from Algorithm 1, it is clear that

- If $\mathcal{A}'$ is given a sample $(\mathbf{G},\mathbf{y})$ with $\mathbf{y} = \mathbf{xG}+\mathbf{e}$, $\mathcal{A}'$ simulates $\mathcal{A}^{\mathsf{Encode}(\mathsf{pk})}(\mathsf{pk})$;
- Otherwise, if $\mathcal{A}'$ is given a sample $(\mathbf{G},\mathbf{y})$ with $\mathbf{y}$ uniform, $\mathcal{A}'$ simulates $\mathcal{A}^{\mathcal{U}}(\mathsf{pk})$.

Hence, $\mathcal{A}'$ solves $\mathsf{McEP}_{q,n,k,\mathcal{F},\eta}$ with advantage $\nu(\lambda)$. □

Lemma 8 (Strong adaptive robustness). *If any code from* $\mathcal{F}$ *can bounded-distance correct* δ *errors efficiently, then the PRC from Algorithm 1 is strongly adaptively robust against any channel introducing a fraction of at most* $\delta' = \delta - 2\eta$ *substitutions.*

Proof. Fix any $(\mathsf{pk} = (\mathbf{G}', \mathbf{o}), \mathsf{sk})$, where $\mathbf{G}' = \mathbf{SGP}^{-1}$, and let $\mathbf{c} \leftarrow \mathsf{Encode}(\mathsf{pk})$. Write $\mathbf{c} = \mathbf{xG}' + \mathbf{e} + \mathbf{o}$. Since $\mathbf{e}$ is an error term introduced by $\mathcal{B}_\eta$, we have $|\mathbf{e}| \leq 2\eta n$ with probability $1 - 2^{-\Omega(n)}$ over the randomness of Encode.

Let $\mathbf{y} = \mathcal{E}(\mathbf{c}, (\mathsf{sk}, \mathsf{pk}))$ for any strong adaptive channel introducing at most a fraction of δ' substitutions, i.e. $d(\mathbf{y}, \mathbf{c}) \leq \delta' n$. Let $\mathbf{y}' = (\mathbf{y} - \mathbf{o})\mathbf{P}$ computed in the first step of Decode. We will now show that $d(\mathbf{y}', \mathbf{xSG}) \leq \delta n$:

$$
\begin{aligned}
d(\mathbf{y}', \mathbf{xSG}) &= d((\mathbf{y} - \mathbf{o})\mathbf{P}, \mathbf{xSG}) \\
&= d(\mathbf{y} - \mathbf{o}, \mathbf{xSGP}^{-1}) \text{ since Hamming dist. is permutation invariant} \\
&= d(\mathbf{y}, \mathbf{xG}' + \mathbf{o}) \\
&\leq d(\mathbf{y}, \mathbf{c}) + d(\mathbf{c}, \mathbf{xG}' + \mathbf{o}) \quad \text{by triangle inequality} \\
&= d(\mathbf{y}, \mathbf{c}) + |\mathbf{c} - \mathbf{xG}' - \mathbf{o}| \\
&= d(\mathbf{y}, \mathbf{c}) + |\mathbf{e}| \\
&\leq (\delta' + 2\eta)n \\
&= \delta n.
\end{aligned}
$$

As a result, $d(\mathbf{y}', \mathcal{C}) \leq \delta n$ so the decoding algorithm of $\mathcal{C}$ will succeed, hence with probability $1 - 2^{-\Omega(n)}$, $\hat{\mathbf{y}} = \mathsf{Decode}_{\mathcal{C}}(\mathbf{y}')$ verifies $\hat{\mathbf{y}} \in \mathcal{C}$ and $|\hat{\mathbf{y}} - \mathbf{y}'| \leq \delta n$, which means $\mathsf{Decode}(\mathsf{sk}, \mathbf{y})$ will return $\top$ with probability $1 - 2^{-\Omega(n)}$, i.e. $1 - \mathsf{negl}(\lambda)$, since n is a polynomial in λ. □

B.3 Proof of Theorem 8

Theorem 8 follows by combining the following Lemmas 9 to 11 in which we demonstrate each individual PRC property of Algorithm 1.

Lemma 9 (Soundness). *For a family $\mathcal{F}$ of linear codes of dimension k, whenever $n(1 - H_q(\delta)) - k = \Omega(n)$, the PRC from Algorithm 2 is sound.*

Proof. The proof of Lemma 6 still holds under the same condition on the decoding limit parameter δ. This is because when $\mathbf{y}$ is a fixed vector, $\mathbf{y}' = (\mathbf{y} - \mathbf{o})\mathbf{P}$ is still a uniformly random vector, even when $\mathbf{P}$ is a distortion matrix instead of a permutation matrix. □

Lemma 10 (Pseudorandomness). *The pseudorandomness of the PRC of Algorithm 2 follows from the hardness of $\mathsf{DistMcEP}_{q,n,k,\mathcal{F},\gamma,\eta}$.*

Proof. The proof is analogous to that of Lemma 7. □

Lemma 11 (Strong adaptive robustness). *If any code from $\mathcal{F}$ can bounded-distance correct δ errors efficiently, then the PRC from Algorithm 2 is strongly adaptively robust against any channel introducing a fraction of at most $\delta' = \frac{\delta}{\gamma} - 2\eta$ substitutions.*

Proof. Let n be polynomial in λ. We will reuse the same proof techniques as in Lemma 8, except that we will crucially use the fact that for a distortion matrix $\mathbf{P}$ of weight γ and for any $\mathbf{c}, \mathbf{d} \in \mathbb{F}_q^n$,

$$d(\mathbf{c}, \mathbf{d}) \leq \gamma \cdot d(\mathbf{c}\mathbf{P}^{-1}, \mathbf{d}\mathbf{P}^{-1}). \tag{2}$$

Fix any $(\mathsf{pk} = (\mathbf{G}', \mathbf{o}), \mathsf{sk})$, where $\mathbf{G}' = \mathbf{S}\mathbf{G}\mathbf{P}^{-1}$, and let $\mathbf{c} \leftarrow \mathsf{Encode}(\mathsf{pk})$. Write $\mathbf{c} = \mathbf{x}\mathbf{G}' + \mathbf{e} + \mathbf{o}$. Since $\mathbf{e}$ is an error term introduced by $\mathcal{B}_\eta$, we have $|\mathbf{e}| \leq 2\eta n$ with probability $1 - 2^{-\Omega(n)}$ over the randomness of Encode.

Let $\mathbf{y} = \mathcal{E}(\mathbf{c}, (\mathsf{sk}, \mathsf{pk}))$ for any strong adaptive channel introducing at most a fraction of δ' substitutions, i.e. $d(\mathbf{y}, \mathbf{c}) \leq \delta' n$. Let $\mathbf{y}' = (\mathbf{y} - \mathbf{o})\mathbf{P}$ computed in the first step of Decode. We will now show that $d(\mathbf{y}', \mathbf{x}\mathbf{S}\mathbf{G}) \leq \delta n$:

$$\begin{aligned}
d(\mathbf{y}', \mathbf{x}\mathbf{S}\mathbf{G}) &= d((\mathbf{y} - \mathbf{o})\mathbf{P}, \mathbf{x}\mathbf{S}\mathbf{G}) \\
&\leq \gamma \cdot d(\mathbf{y} - \mathbf{o}, \mathbf{x}\mathbf{S}\mathbf{G}\mathbf{P}^{-1}) \quad \text{by Equation (2)} \\
&= \gamma \cdot d(\mathbf{y}, \mathbf{x}\mathbf{G}' + \mathbf{o}) \\
&\leq \gamma \cdot (d(\mathbf{y}, \mathbf{c}) + d(\mathbf{c}, \mathbf{x}\mathbf{G}' + \mathbf{o})) \quad \text{by triangle inequality} \\
&= \gamma \cdot (d(\mathbf{y}, \mathbf{c}) + |\mathbf{c} - \mathbf{x}\mathbf{G}' - \mathbf{o}|) \\
&= \gamma \cdot (d(\mathbf{y}, \mathbf{c}) + |\mathbf{e}|) \\
&\leq \gamma \cdot (\delta' + 2\eta)n \\
&= \delta n.
\end{aligned}$$

As a result, $d(\mathbf{y}', \mathcal{C}) \leq \delta n$ so with the same reasoning as in Lemma 8, $\mathsf{Decode}(\mathsf{sk}, \mathbf{y})$ will return $\top$ with probability $1 - \mathsf{negl}(n)$, as desired (recall that n is polynomial in λ). □